mathNEWS FREE

VOLUME 148 • ISSUE 6

APRIL 1, 2022

WANT THE REST OF THIS COVER?

SUBSCRIBE TO mathNEWS++ TO GET UNLIMITED ACCESS TO ALL PAST, PRESENT, AND FUTURE COVERS!

ONLY $7720.25 PER ACADEMIC TERM

CC SOME RIGHTS RESERVED

9 770705 041004

mastHEAD

"mathNEWS'S NEXT MONEYMAKING VENTURE SHOULD BE...?"

Hi everyone! First, housekeeping: with this being the long-awaited inaugural issue of **mathNEWS**++, keep in mind that the **mathNEWS** issue you're reading right now is the free version, **mathNEWS**[FREE]. We're still working on the formatting here, but in the future, please refer to anything published by **mathNEWS** as being either **mathNEWS**++ or **mathNEWS**[FREE]; there are no issues of just '**mathNEWS**'.

Next, some shipping information: if you've made a pre-order before March 14, you should be getting your issue delivered by the end of today (check for broken windows in your apartment). For those of you who are only reading the free version, continue as normal.

To support a smooth transition, we're going to gradually move more content to **mathNEWS**++ over time, so there is still a lot of premium content in this issue. I particularly enjoyed kai's *ALBUM REVIEW: KANYE WEST, 2016 TO PRESENT* and boldblazer's *Glorious Canadian Soccer*, both of which got me excited about and interested in things that I previously knew nothing about.

You'll also find many N-things-list articles this issue, covering a breadth of topics from diagnosing computer hardware problems to reminiscing over a long friendship. If you particularly enjoy any of these articles, please remember that the other half of these N things can be found in **mathNEWS**++!

Finally, several monumental series hit concluding notes this issue: if you're a fan of C++, Brainfuck, or Gaming, be excited. Oh, and there's something like four songs this issue too, two of which have the same title? Exciting times.

bipED
Editor, **mathNEWS**

ARTICLE OF THE ISSUE

This issue's Article of the Issue goes to *Cucumbers*, by aphf, overcoming many formidable challengers. Come to MC 3030 for your $25, and check your windows.

bipED
Editor, **mathNEWS**

Xx_crypto_lover_69_xX	HODL MATHNFTEWS TO THE MOON!!! DIAMOND HANDS!1!! GET ALL YOUR FRIENDS IN NOW!!! FOR THE LOW LOW PRICE OF 5 ETHER!!!!
SecretSquirrel	Goosemas day merchandise
Fluffiest cactus	A cookbook, unironically
Deriving for Dick	WealthSimple **mathNEWS**: The fastest and simplest way to write a **mathNEWS** article.
jeff	Sink all of **mathNEWS**'s funding into venture capital and declare insolvency.
hu	Charging printers for exclusivity rights
mouse	Charging readers to unsubscribe from **mathNEWS**++
Skit	Committing fraud.
someBODY	Stealing the Hope Diamond (**mathNEWS** field trip?)
cutlet	**uberREADS**: add a copy of **mathNEWS** to any Uber Eats delivery for just $1.99 (tax not included)
Wink wonk	Onlymaths – Onlyfans but math (obviously)
x	Onlyfans
Xx_420SonicFan69_xX	Extortion and blackmail (using pizza of course)
evilevievil	More merch: **mathNEWS** bunny plushie
A cool pen name	Invest in UGO II.
sqrt(cause)	Stealing the Imprint fee.
Abald Man	A money printer.
Golden	Organise a militia and reclaim the wealth from the bourgeoisie.
whoknows	Hire an accountant to play with the numbers
loop	Hardcover **mathNEWS** yearbook with Pizza coupons stuck in the pages (expired ones of course).
methNEWS	Invent math 2 (or NEWS 2)
str(n)cat	**mathNUDES**
Finchey	Auctioning off the vintage, MISB (mint in sealed bottle) Orbitz in the office.
terrifiED	Cut spending, hike the **mathNEWS** fee, and keep asking for grants and donations — which is pretty much what every government does these days, so at least we'll be in pretty good shape for eventual world domination.
caffeinatED	**mathNEWS** maid café
clarifiED	**mathNEWS**++. Already there.
enamouredED	Gamer game stream ✨

mathNEWS++ contributor gala RSVP coming soon.

YANG ZHONG, mathNEWS EDITOR FOR WINTER 2022
ALONG WITH CHEN CHAI, TERRY CHEN, NAMAN SOOD, AND CLARA XI

mathASKS 148.6

FEATURING PROFESSOR NAOMI NISHIMURA

ABALD MAN: CAN YOU EXPLAIN YOUR THESIS IN TERMS A THIRD-YEAR MATH STUDENT COULD UNDERSTAND?

Yes, but my current work is much more interesting.

My thesis was an attempt to bridge the gap between the theory and practice of parallel computing. Theoreticians were coming up with clever algorithms based on the unrealistic assumption that multiple processors could communicate by writing simultaneously to shared memory. The idea of coming up with an asynchronous model made so much sense that two other grad students and I all simultaneously came up with similar ideas. We all presented our models at a lovely conference in Greece, which, it so happened, turned out to be the moment at which interest in our work peaked.

Most of the work I do now is in one of two areas, both of which focus on trying to find nice algorithms for hard problems. (Although you don't need to know the technical details, here "nice" means having a worst-case running time that is polynomial in the size of the input, and "hard" means NP-complete, that is, unlikely to be able to be solved using a nice algorithm.)

Parameterized complexity looks at hard problems and tries to identify features of the problem or input, known as parameters, to "blame" for the hardness. Just like you can categorize problems as being easy or hard (in P or NP-complete) in the conventional setting, you can categorize problems as being easy or hard in the parameterized setting. A parameterized problem is easy if you can create a sort of "sliding scale" algorithm, where the algorithm is nice when the parameter size is small, but not so nice otherwise. (In case you are interested in the details, such an algorithm has worst-case running time that is the product of a polynomial in the size of the input and some potentially really terrible function on the size of the parameters.)

Reconfiguration problems focus on the difficulty of transforming one solution into another rather than the difficulty of solving the problem in the first place. For a simple type of modification, the goal is to make step-by-step changes from one solution to another such that the result of each modification is also a solution. There are problems that are easy to solve but hard to reconfigure, and problems that are hard to solve but easy to reconfigure. We don't yet know why.

XX_420SONICFAN69_XX: HOW DID YOU NARROW YOUR FOCUS OF WHAT YOU WANTED TO PURSUE IN MATH INTO ALGORITHMS AND GRAPHS SPECIFICALLY?

I didn't deliberately narrow my focus. It was only through luck and the kindness of strangers that I stumbled into my current area of research.

After high school, I chose to attend Yale, in part because I didn't have to declare my major until the end of my second year. Even then, instead of choosing between math and classics, I ended up majoring in both.

By the time I graduated from Yale, I knew how to be a student, but not what to do next. Most of my friends were busy interviewing for jobs. One had applied to teach English in China, and suggested that I do the same. I figured it was a good opportunity to practice being interviewed. By the time I had convinced my interviewers that I wanted to see how much I liked teaching before committing to grad school, I had also convinced myself!

After two years in China, I knew that I wanted to teach, but not what I wanted to teach. Even though I had only taken a few courses in computer science at Yale, I was attracted by the job prospects and the promise of graduate schools providing full funding.

In grad school I was exposed to graph theory and combinatorics for the first time, but since my supervisor was interested in parallel computation, graph algorithms didn't become my main area until after I graduated. I had a friend in grad school with whom I enjoyed discussing problems, and since he was interested in graph minors, looking at graphs of bounded treewidth was a natural choice. A few years later I advertised for a postdoc, and happened to hire someone with expertise in parameterized algorithms. Years later, he had a friend who was visiting Waterloo and suggested that the friend look me up. After a few years of working together, that friend starting working on reconfiguration problems, and introduced me to the area.

In retrospect, it is now obvious to me that my brain naturally gravitates to trying to figure out structure and patterns and then put that knowledge to use. It explains why I like learning grammar in languages, and the attraction of music and dance. In the process, though, I was just working on interesting problems with friends.

LOOL: WHAT'S YOUR SECOND FAVOURITE AREA BESIDES PARAMETERIZED ALGORITHMS AND RECONFIGURATION PROBLEMS?

My favourite research areas involve finding structure. Some, but not all, use parameterized algorithms, and some, but not all, use reconfiguration problems. Two of my most-cited papers use neither.

EVILEVIEVIL: WHAT IS YOUR FAVOURITE RESULT IN COMBINATORIAL RECONFIGURATION THAT MAY BE SURPRISING/COUNTER-INTUITIVE TO AN AVERAGE MATH STUDENT LIKE ME?

What has surprised me the most is that reconfiguration has turned out to be useful in real life. I knew it was an attractive area, and a great way for students to start research, but possible real-world applications seemed a bit far-fetched.

Reconfiguration is now being used for efficient power distribution in Japan, and can be used in quantum simulation.

WINK WONK: WHAT PROBLEM DO YOU FIND MOST INTERESTING / WOULD LIKE TO SEE SOLVED IN THE NEAR FUTURE?

The problems I care about most are not research problems, but societal problems of equity, opportunity, and justice. Although I have been able to help build an inclusive and supportive research community, I have no illusion that my research itself is going to make the world a better place.

I'd like everyone to have a fair shot at learning computer science, whether or not they have the privilege to grow up with access to fancy equipment or role models. With that aim, I've developed free courseware sponsored by the Centre for Education in Mathematics and Computing; developed and taught courses for non-majors (CS 115, CS 231, and CS 234), for high school math teachers, and for graduate students at the African Institute for Mathematical Sciences; and served as mentor in a program at a reserve in Northern Manitoba.

Long after I forget all the research I've done, I'll remember the students who went from fearing computer science to loving it enough to change their programs of study. I'll never tire of seeing the look in a student's eyes when they go from not understanding to understanding, the "light switch" moment when everything clicks.

BOLDBLAZER: IT'S BEEN A LONG TIME SINCE THE FEATURED PROF WAS SOMEONE WHOSE COURSE I'VE TAKEN, IN THIS CASE CS 234 LIKE 2 YEARS AGO. HAS THE COURSE CHANGED MUCH SINCE THEN?

Even before the pandemic, I was constantly tinkering with my courses. That process accelerated over the last two years, since I figured I might as well get started on the online versions of CS 231 and CS 234 that I'd promised to (eventually) develop.

In 2020, I made videos out of all my lectures, tried to translate in-class questions into quizzes and exercises, and started discussions with the Centre for Extended Learning about plans for polished courses with full accessibility (e.g. closed captioning for videos and alternative text for images).

In 2021, I taught both courses using Open edX, interspersing videos and text with a variety of other features such as multiple choice questions and editable, executable coding sections. At my request, the site was enhanced to allow you to hover over a term to see its definition and enter your own answers to exercises before revealing suggested solutions.

I hope I've brought both courses up a few notches, not just because of all the bells and whistles, but also because of the improvements in my presentation of material resulting from my having to think about it in a new way.

EVILEVIEVIL: WHAT IS YOUR FAVOURITE GRAPH AND WHY?

The tree. It's the perfect gateway graph for those new to thinking about problems in a structural way: trees naturally model real-life data that most people will have encountered, and trees lend themselves well to demonstrating a variety of algorithmic paradigms. Trees are also complex enough to allow for nontrivial properties and algorithms.

PREDAP: WHAT'S YOUR FAVOURITE THING ABOUT WATERLOO?

The students. I chose to come to Waterloo because of its attempt to strike a balance between being research-focused and teaching-focused and due to the reputation of the students, including the high percentage who are the first ones in their families to attend university. The research-teaching balance has gone off-kilter from time to time, but the students have never disappointed me. On the whole, my classes are made up of curious, open-minded learners, who approach challenges with a willingness to work rather than a sense of entitlement.

SKIT: HOW COME EVERY TIME I ASK A PROFESSOR WHAT THEIR FUNNIEST STORY IS, THE ANSWER NEVER GETS PUBLISHED?

APHF: WHAT IS YOUR LEAST FUNNY STORY?

I'll let you decide how to categorize a story about a time I had to suppress my desire to burst out laughing.

When I was a grad student, I was given the opportunity to teach a course on computer literacy, which at that time entailed explaining such terms as "software", "hardware", "keyboard", and "monitor". Since the course didn't assume any background in anything, the most challenging topic was an introduction to binary numbers.

One day a student came to office hours, and asked how to translate a number into binary. I carefully went through each step, explaining how I was figuring it out, and encouraging questions. He had no questions, but as soon as I was done, asked for a second number to be translated into binary. I tried to get him to figure out some of the steps on his own, but he seemed more interested in looking at his notes than in listening to me. When he asked for a third number, I suggested that I observe his work this time.

It was only at that point that he admitted that he wasn't really there to learn, but to test me. There had been an error in his notes, and he needed to determine whether it was because he had transcribed the information incorrectly or it was because I didn't know the material.

CLARIFIED: WHAT IS YOUR FAVOURITE FOOD?

Garlic, both to eat and to grow. It requires so little space and so little attention that even an unskilled gardener like me can grow a year's supply—plus enough extra to plant for the next year—in a community garden plot.

πLLOW PRINCESS: WHERE IS YOUR FAVOURITE TOILET ON CAMPUS?

I don't have a favourite toilet, but I admit a fondness for the communal foot-operated hand-washing stations in MC.

ADVICE FROM A CONTRARIAN

prof**THOUGHTS** *148.6*

My life is a demonstration of what happens if you are too unconventional to pay attention to conventional wisdom: I didn't wait until tenure to have children, I didn't let work get in the way of a good night's sleep, I took vacations every year, and I didn't put research before teaching.

Not surprisingly, I'm not rich and famous, but that was never my goal in the first place. (My biggest claim to fame is being the first CS professor at UW to become a grandmother.) As my reward for ignoring conventional wisdom, conventional wisdom has also ignored me: I don't struggle with work-life balance, I didn't make my best contributions before turning 30, and I don't suffer from burnout or stress. Life is good.

Even if your goal is fame and fortune, maybe some of my unconventional advice will help you.

1. DON'T BELIEVE IN YOURSELF.

I mean that the you of today shouldn't trust the you of yesterday.

If you did a bunch of work on a project or a proof yesterday, today you'll want to start building on what you did. That's natural enough, but I recommend against it. Instead, view yesterday's accomplishments with suspicion.

Particularly dangerous is when you've been supplied with a statement to either prove or disprove. Typically, the more time you've invested in trying to prove that it is true, the less likely you are to consider that it might be false. You don't want to admit that you might have wasted time by making the wrong choice, so you double down. (Yes, I used to do this, too.)

I suggest instead switching back and forth between time spent on proving the statement and time spent on disproving the statement. The difficulties you encounter in one approach will help you in the opposite approach, and you won't get overly invested in one (possibly incorrect) outcome.

I have one friend who is particularly skilled at finding counterexamples. The rest of us will be chattering happily about our research ideas, leaping forward to the next conclusion, while he sits quietly and takes it in. When he opens his mouth, our hearts sink, as he invariably has found the flaws in our thinking. Friends like that are precious.

2. QUIT WHILE YOU'RE HAVING FUN.

Solving problems is a ride on an emotional rollercoaster, where you go from the elation of a good idea to the despair of finding an error. I used to stop working when I got stuck. That turned out to be a terrible idea, since I would associate the problem with failure, and dread going back to it.

I eventually learned that it is much better to quit when I'm feeling good about my achievements, even if they are small and even if they are illusory. I use this method often when working collaboratively with others. We will stop our work session when we have ideas that we're excited about exploring. In the following work session, even if we realize that they are bad ideas, we still have enough time and energy to stop at the next "peak."

3. DON'T GO IT ALONE.

One year, one of my students got into an accident, and was struggling to complete work while getting around on crutches. Until I suggested it, he hadn't considered contacting AccessAbility Services, as he didn't think it was designed for people like him. Fortunately, he was able to change his way of thinking about himself, get help, and thrive.

Unfortunately, I've been less successful in convincing students that attending office hours is not a sign of weakness. I have lost count of the number of times that students have approached me after a midterm, asking for recommendations on private tutors to hire. My answer is always the same: Why not use the resources that you've already paid for, attending office hours with course personnel who were chosen for their expertise in the subject?

As a bonus, if you ever need to ask a professor for a recommendation letter, they are much more likely to say yes, and much more likely to be able to write a strong letter, if they have actually had one-on-one interactions with you.

Although it might sound ridiculous, I also recommend that you hold your own office hours and come up with your own exercises for friends to try. You'll get extra practice in your written and oral communication skills, which are essential for just about every job, and you're likely to find that in the process of teaching material to someone else, you'll understand it better.

(You might also like it better. I hated formal languages in grad school, but CS 360 is now one of my favourite courses to teach.)

4. DON'T DRESS FOR SUCCESS.

I hope that the world has changed from the days in which my friend chose to interpret "those shoes look comfortable" as a compliment from her boss, subsequently was fired for her choice of footwear. Judging from the prevalence of foot deformities among women my age, she was clearly not the only one under such pressure.

Yes, I know I should be taking this up with employers, not employees. But you may be employers some day, so please keep these words in mind.

5. KEEP YOUR EYES OFF THE PRIZE.

I once met a friend of a friend whose career goal was to become vice-president at a company. (Is that even a thing?) He had all the steps planned out, and explained step by step how many years he'd spend in each position, climbing ever upwards. He had no particular companies in mind, nor any anticipation of enjoying his work, just a clear idea of what title he wanted each year of his life.

If that's the way you can keep yourself motivated to do what you do, go for it. For the rest of us, staying on track only happens if we can enjoy the process along the way. Most of our time is spent on the journey, making it arguably more important than the destination.

Another problem with being focused on a specific goal is that one can become distracted from reassessing whether or not the goal is really the right one. Which leads to …

6. BE A QUITTER.

I was very lucky to have a role model in quitting. When I was in high school, my father made a pivot from theoretical physics to patent law. It wasn't easy, as he was working full time, studying by correspondence during the evenings and weekends, and for his first job he took a huge pay cut. In time, though, it was clearly the right decision: he had a long and rewarding career, retiring only when he turned 85.

Many students arrive at UW with a particular plan in mind, and then, upon learning more about themselves or about what the area is like, realize that they have made the wrong decision. I made the same mistake in university; once I realized that what appealed to me in high school Latin class didn't translate well to a major in classics, I should have cut my losses instead of completing the major for the sake of finishing what I started.

And don't blame yourself for having made the wrong choice in the first place. It makes no sense that you're being asked to choose your future path at an age at which your brain hasn't completed its development, and with little knowledge about what various options entail. It is a miracle that anyone is able to follow through with plans made by their 16-year-old self, who knows so little about themselves or the world.

7. STICK WITH WHAT YOU'RE BAD AT.

Most of us get praise for our achievements — whether by parents, friends, or teachers — and then start to focus on what comes most easily and naturally to us. Along the way, we often conflate excellence and enjoyment, forgetting that skill and pleasure need not be correlated.

Here are a few reasons that you lose out by limiting yourself to what you're good at:

- You're missing out on activities that can give you a lot of joy. I know this first-hand, from waiting until I was in my 50s before deciding to ignore the naysayers and finally take singing lessons.
- You're missing the chance to become better at what you're bad at. Although we expect practice to be necessary to get our bodies to do what we want, such as in sports or a musical instrument, we often forget that the same is true for our minds.
- You're missing an opportunity to understand others. If you remind yourself what it is like to struggle, you will have more empathy for those who struggle with areas that come easily to you, like math.

And if you're good at everything you do, try something new so that you can follow this suggestion.

Naomi Nishimura

EASY MATH PROBLEM FOR BABIES ($500 PRIZE FOR THE SMARTEST BABY)

Define:

$$H_n := \sum_{k=1}^{n} \frac{1}{n}$$

Prove for all $n \in \mathbb{N}$,

$$\sum_{d|n} d \leq H_n + e^{H_n} \ln(H_n)$$

To the first person to solve this and send a correct solution to legitmilleniumprizewinner@gmail.com, I will personally give $500 (this is not a joke).

Adam Jelinsky

WHAT YOUR MATHSOC VPA IS DOING: PART 6

Hello! My name is Vincent, and this term I am your MathSoc Vice President, Academic.

I've been updating students in **mathNEWS** every issue about what I've been doing at MathSoc to improve the undergraduate experience in the Faculty.

So here's the highlights since last issue.

PROCTORING SOFTWARE

Most of my time these past two weeks was spent on this. It paid off.

After over 1200 signatures to MathSoc's petition, a 15 page rationale, and some speeches, the Math Faculty Undergraduate Affairs Committee adopted the following motion:

> *Undergraduate Affairs Committee recommends that Faculty Council adopt the following resolution:*
>
> *Faculty Council prohibits the use of Proctoring Software for tests, quizzes, or examinations in any Undergraduate Course owned by the Faculty of Mathematics. This includes courses offered on campus, online, or in hybrid delivery mode. This prohibition will take effect at the start of the Spring 2022 term and will remain in effect until and unless it is repealed by Faculty Council. The Faculty will seek agreement with the Faculty of Engineering and the Provost to extend this prohibition to Software Engineering courses. This prohibition on Proctoring Software shall not apply in cases where an external accreditation body requires the use of Proctoring Software.*
>
> *For clarity, "Proctoring Software" includes but is not limited to tools such as ProctorU, Proctorio, Proctortrack, Respondus LockDown Browser, and ExamSoft. "Proctoring Software" does not include general-purpose video-conferencing software such as Zoom or Microsoft Teams.*
>
> *UAC may approve exceptions for the experimental use of new innovations in the area of proctoring software.*

Please note that this motion has not yet fully passed. It still needs to be considered by Faculty Council on April 19th. However, passing through the Undergraduate Affairs Committee is the last step towards passing at Faculty Council. You can still sign MathSoc's petition at https://bit.ly/proctorpetition or scan the QR code below:

PD AND WORK TERM REPORTS

Continue to stay tuned I guess. The meeting where the big change will be decided is on April 25th, so it will probably be in the first issue of next term that I announce what happened in **mathNEWS**. MathSoc will be sure to announce in our other communication channels though as soon as the decision is made.

Vincent Macri
Winter 2022 MathSoc Vice President, Academic

WHY DID I SIGN UP FOR A PODCASTING COURSE?

If you are anything like the people I talk to on a regular basis, then pandemic burnout is real. It's just very hard to find anything to get excited about as it feels like our lives have been put on pause for the last 2 years. To try and get myself out of the pandemic blahs, I thought that maybe if I sign up for a course, this will force me to actually commit to a thing.

Past me was hopelessly naive. As soon as I saw the amount of video tutorials in the course, my motivation to get through the material shrivelled. I am very much of the opinion that most video tutorials would be better off as an article. It would be faster to read and you would not have to pause the video to revisit the part you are interested in.

So now instead of actually creating a podcast, I just gave up at the slightest obstacle and let life get in the way, leaving me with the guilt of once again wussing out on a thing. I suppose a healthier outlook would be that I am not yet successful, and that in fact I can return to this endeavour instead of thinking in an incredibly binary manner of failure or success. But why would I be kind to myself? Why prioritize mental health when you can instead level up your skill in physic damage?

I suppose I could approach the problem from a different angle and recruit people to help with the podcast and hold me accountable. That requires finding people. Surely it could not be as easy as talking to **mathNEWS** folks who are always talking about creating a podcast, and having an actual plan to follow through. That sounds hard.

Incidentally about following through on things: I finished this article because I put 20 minutes of creative endeavour on my daily to-do list, and this was the easiest way to honour that goal. So instead of dwelling on efforts that haven't born fruit, I will choose to focus on the success I did achieve.

Beyond Meta

REAL OR NOT REAL?

Sometimes I doubt that I am real. It's like, if I die, who will know I even existed? Eventually, no one will remember me, and all my life, everything that was driving me, my dreams, my fears, the secrets I never told anyone, the love I had, the chances I missed, the scars both on my mind and my skin, everything that makes me myself, everything will disappear. Like it never existed.

It's like the story of the tree, that is so deep into the forest that no one ever came, and no one will ever come. When it falls, will it make a sound? Even if it does, does it matter? No one ever thinks about this tree, about those things that we ignore every day because we don't know they're here. Thinking of all the opportunities I missed, all the friends I didn't make, it's enough to make me feel dizzy.

Just so I'm clear, it is not such a depressing thought. Eventually, no one will be remembered. And even I missed countless opportunities that could lead to countless parallel worlds, I'm quite happy with my life.

From time to time, I listen to the trees. Are they real? If I believe they are, maybe then I am real as well. Maybe it wasn't all a dream. Maybe I didn't imagine it all.

PhilosophicalSoul

TOP 10 WATCHUGO VIDEOS OF ALL TIME

8. Top 10 Ω-3 fatty acids of All Time
2. Top 10 Lithuanian household gods of All Time
6. Top 10 Judete in Romania by population of All Time
3. Top 3 Characters of the Faerie Wars series of All Time
10. Top 10 Football clubs in Malta of All Time
7. Top 10 Logical Symbols of All Time
4. Top 10 Canadian incumbents of All Time
9. Top 10 Pakistani Punjabi language films of All Time
5. Top 10 Law enforcement agencies in Germany of All Time
1. Top 1 English words of Manx origin of All Time

UgoFan69

You can watch these and other such videos at https://www.youtube.com/channel/UCVjOY5-nw_G14JSgOFrrMvA.

UNLOCK THIS FILLER WITH mathNEWS++!

SEE FACING PAGE FOR DETAILS

N mathNEWS++ FEATURES YOU WON'T SEE IN mathNEWS

- (Mostly) backwards compatibility with **mathNEWS**
- More angle brackets
- Classes with methods
- Object-oriented principles
- Beautiful, expressive syntax
- An extensive standard library
- Move semantics (since **mathNEWS**++11)
- Templates

jeff

N THINGS TO DO WHEN YOUR COMPUTER DOES NOT POST

Recently during an overclocking attempt, I set my memory controller voltage to 1.19V and the computer failed to POST (show the boot logo screen) after doing so, even after removing the CMOS battery. Thankfully, it turns out that some BIOS settings are stored in NVRAM (meaning that removing the battery does not clear everything), and I did not just fry my $100 motherboard and $300 CPU. To celebrate me not destroying $400 of hardware, I made a list of N things to do when your computer refuses to POST:

- Make sure your parts are compatible with each other; do your research
- Make sure your parts are installed correctly
- Remove your CMOS battery and disconnect your power cord, and wait for a few seconds before reinstalling battery and power cord
- Reset your CMOS using the CMOS reset header (or button on your motherboard), consult your motherboard's manual for step-by-step instructions on how to do this
- If your motherboard has a buzzer or an 8-segment display, look up the error codes (or any other abnormal behaviour)
- Ask a senpai or a more knowledgeable friend for help
- Test each component piece-by-piece in a known good system
- Cry *[Editor's note: I can vouch for this one being extremely effective]*

tokyocatboy

DECADE 🤩

Did you know I'm new to eye 👀 makeup?! It's so fun and unique 🤩 and quirky ✨!!! I 💝 love 💝 eye makeup!!! Like putting on a costume 🎭! Sometimes I see someone else with eye 👀 makeup on and it's so pretty and fantastic 😍!! I think they might have been doing eye 👀 makeup since they were in middle school 🤭!! That's one DECADE 🤓 of eye 👀 makeup experience!! Wow!!!! Maybe someday I'll have one decade of eye 👀 makeup experience and be as good as them too ☺

enamourED

DREAM OVERFLOWED

If you still remember those pop culture references from 10 years ago, you should be able to recall something something *Inception* something. If you don't, I'll briefly bring up the main ideas: Assume you can dream in your dream. Assume that your brain won't fry because you dream too deep (according to the movie, if things go super wrong you will fall into "limbo", which is like the 4^{th} or 5^{th} layer of dream). We can formalize the ideas behind *Inception* into the following statements: If we treat each layer of the dream like a function call, and each layer of the dream takes up a constant portion of each person's brain storage/stack frame, then in the worst case, the brain will completely fall into "limbo" (in which case, the dream has used up 100% of one's brain and the dreamer has lost any trace back to reality). And I can guarantee you that things can go way worse than "limbo".

It was a peaceful Friday evening, and I was somehow having a dream, which doesn't usually happen to me these days. To cut to the point, it wasn't a sweet dream. I woke up screaming in my head and sweating all over. Just those kinds of nightmare I used to have when I was 7. I opened my eyes real quick, and weird things happened. I was 10 seconds ahead of the place where I should've woken up. Having written a lot of loops in my CS class these days, I quickly realized what happened: I had somehow looped inside a nightmare. And this real horrifying thought woke me straight up, thank God to reality.

What went wrong here? Well, I can't exactly remember the things that scared me up, but one thing's for sure: I got lazy and tried to construct a dream in my dream with the same dream space I was currently dreaming in, and when it came time to wake up I messed up which layer of dreams I was dreaming in, i.e. I couldn't tell if the dream was on the bottom of the dream stack where I should wake up to reality, or if there deeper layers beneath me. The consequence was that I kept waking up to the same dream I was running away with, while at the same time probably overflowing my brain stack and forcing my subconscious to reboot and force me out.

My point is: you beter dream creatively, or else things can go real bad.

Loop

THE FIVE WAYS THAT MATH STUDENTS DRESS

1. PROBABLY HAS IMPOSTOR SYNDROME

- owns a turtle neck
- wears anything but jeans (khakis, leggings, palazzo pant)
- some type of boot, like a chelsea boot

2. NEVER LEARNED TO DRESS

- a t shirt, graphic tees mostly
- a full zip hoodie, left unzipped
- sweatpants or jeans
- running shoes
- all of these clothes are Christmas presents

3. CRAZY RICH INTERNATIONAL STUDENT

- disposable mask, but not blue ones
- trench coat
- designer bag
- maybe a Doc, maybe a converse

4. COMFORTABLE AND CASUAL

- exclusively wears H&M or Uniqlo
- owns a cardigan
- wide legged jeans
- colors are earthy or monochromatic

5. WANNABE ENGINEER

- Waterloo branded hoodie
- jeans, exclusively

Deriving for Dick

YOU HAVE REACHED YOUR DAILY blackBOX LIMIT.

SUBSCRIBE TO READ MORE blackBOXES WITH mathNEWS++ TODAY!

ONLY €666.66 PER EXACT SOLAR YEAR

EASY FIRST-CLASS FUNCTIONS IN C++ FOR EVERYONE

ANGUISH. TORMENT. DESPAIR. C++.

Now for a tougher, *less* approachable problem. Recall *C++ Lambdas Aren't Closures, Until They Are* from **mathNEWS** 148.3, wherein terrifiED describes why C++ lambdas are, in fact, first-class functions. That's nice and all, but this still kind of sucks. Why? Well, you should now recall *my* article from 148.3, *III, Lambdas? Oh, You Mean Functors?,* wherein I describe how lambdas are actually just instances of functors (i.e. classes with an **`operator()`** method) generated at compile time. In particular, these classes are generally distinct for each lambda expression appearing in the program. It's impossible to determine whether two programs are equivalent, and the same goes for lambda expression, so the compiler plays it safe and gives them all their own classes (they also have their own variable copies that we probably don't want them to share). Since the class name is generated by the compiler, we just have to use `auto` when declaring the lambda, and just let the compiler deduce the type after generating the functor. This is problematic for us as first-class function enjoyers, because it means that two lambdas with the same parameter types and return type will have different types, whereas two such distinct regular old functions *would* have the same return type (e.g. two functions which take two `floats` and return an **`int`** would both have type **`int(float,float)`**.

This is particularly problematic when we want to write functions that take lambda objects as parameters. We could do something like this:

```
template<typename F>
void foo(F fun);
```

OR, equivalently but more succinctly, since C++20 we can use abbreviated function templates, i.e.,

```
void foo(auto fun);
```

and this *works*, I guess, but it's really not ideal. That **`auto`** will accept literally any type. It doesn't place any constraints on what *kind* of lambda is given/what the signature looks like, and worse, it doesn't even guarantee that the argument is even a function! If we do things this way, then we're trusting whoever uses this function to use it right. I mean, fine, I hear you C++20 fans out there screaming about concepts and constraints. Fine, let's go through the motions and see just how wrong you are:

```
void foo(auto fun) requires requires
    { {(int(*)(float,float))fun}; } { /* ... */ }
// ...

auto bin = [](int x, int y) → int { /* ... */ };

auto bar = [](float x, float y) → int { /* ... */ };
foo(bin); // won't compile
foo(bar); // will compile!
```

Great! It works! Problem solved! If only it were that simple. Allow me to nudge things just a little:

```
void foo(auto fun) requires requires
  { {(int(*)(float,float))fun}; } { /* ... */ }
// ...
int z;
auto bin = [z](int x, int y) → int { /* ... */ };
auto bar = [z](float x, float y) → int { /* ... */ };
foo(bin); // won't compile
foo(bar); // won't compile either!?
```

Mmm. Right; so perhaps you see. If our lambda captures, then in fact, we cannot cast it to a function pointer type. It won't work. The best we could do, then, might be something like this:

```
void foo(auto fun) requires requires
  { {fun(3.5f,3.5f)} → std::same_as<int>; }
```

Sure, that's fine and all, but then `foo(bin)` would compile, because the **`float`** arguments would get casted into **`int`**s. So, we have some level of certainty on our return type, but we lose all guarantees on the parameter types.

We're this far in and we've only gone over approaches that don't work. I can hear you now: "C++ bad language! Use Haskell instead! Fuck you!", et cetera. Worry not, friends: the game is not over yet. What if there existed some uniform interface such that we could wrap lambdas with equivalent signatures under one common type? Could *we* do something like this? Let's find out.

Alright, suppose this "wrapper" class is called `Function`. The interface we're going for is something like this:

```
Function<int(float,float)> F = bar;
Function G{[=](double,int) → int { return 3; }};
int p = F(3.5f, 2.6f);
int q = G(3.14, 5);
```

It'd also be nice if we could use this class to wrap regular old functions too, so let's try to keep support for that in mind. Let's start on the basics.

```
template<typename> class Function;

template<typename Ret, typename... Params>
class Function<Ret(Params...)> { /* ??? */ };
```

How should we wrap the actual underlying lambda? This is a little tough. We can't directly use `Ret` or `Params` here; those probably won't even show up in the functor type. We want to figure out the type based on what's passed into the constructor, but we also need to give a specific type for the member to store this. Maybe we could have some layer of indirection with the *actual* lambda getting stored inside

some other object, and resolve some method to call based on a pointer type… hmm. It's hard for me to motivate this naturally, it's just sort of an idea you need to get. But the point is, we're finally going to put aside some of our compile-time glory and acquiesce to a little bit of dynamic, run-time decision-making. You heard right: I'm talking about **dynamic polymorphism.** That nonsense you would've learned about in second-year.

The idea is that we're going to create an abstract `Callable_Base` template class taking `Ret` and `Params`, and store a `Callable_Base<Ret(Params...)>*` in our `Function`. For now, this abstract class will just have a pure virtual **`operator()`**. Further, we'll have a concrete `Callable` template class which takes `Ret` and `Params`, but *also* takes a type `F` representing our lambda type. A `Callable<F,Ret(Params...)>` will own an `F fun`, and will override the base class **`operator()`** method to return `fun` evaluated with some given parameters. We'll make our `Function`'s **`operator()`** call the `Callable::operator()` polymorphically; and finally, how will we let our constructor accept any lambda type? Well, to hell with it, let's make our constructor a template too!

Right, okay, I'm excited now. Let's put that all together:

```
template<typename>
struct Callable_Base;

template<typename Ret, typename... Params>
struct Callable_Base<Ret(Params...)> {
  virtual Ret operator()(Params... params) = 0;
  virtual ~Callable_Base() {}
};

template<typename,typename>
struct Callable;

template<typename F, typename Ret, typename... Params>
struct Callable<F,Ret(Params...)> : public Callable_
Base<Ret(Params...)> {
  F fun;
  public:
    Callable(F fun) : fun{fun} {}
    Callable(const Callable& other) : fun{other.fun} {}
    Ret operator()(Params... params) override {
      return fun(params...);
    }
};

template<typename Ret, typename... Params>
class Function<Ret(Params...)> {
  Callable_Base<Ret(Params...)>* c;
  public:
    Function(std::nullptr_t) : callable{nullptr} {}
    template<typename F>
    Function(F fun) : callable{new Callable<F,Ret(Params...
)>{fun}} {}
    ~Function() { delete callable; }
    Ret operator()(Params... params) {
      if (callable) return c->operator()(params...);
      else throw std::string{"called null function"};
    }
};
```

This seems pretty thorough, right? And, in fact, it'll just about work! We can now do something like `Function<int(float, float)> F = bar`, and it'll work; we'll be able to do `F(0.0f, 3.14f)` and we'll get an **`int`** back! It makes a copy of the lambda with `new` too, so we can lose the original lambda `bar` and still still use `F`.

Perfect then! Let's go on our merry way and start passing these around. Here I go:

```
void awful(Function<int(float,float)> fun) { /* ... */ }
// ...
int main() {
  int z = 5;
  auto bar = [z](float x, float y) -> int { /* ... */ };
  Function<int(float,float)> F{bar};
  awful(F);
  return F(3.5,4.3);
}
```

It compiles, great. Now I'm going to run it:

```
Segmentation fault (core dumped)
```

Marvelous. In our fit of hubris, we forgot about copying. Our `F` gets passed by value, hence copied, into `awful`, and we're using the compiler-provided copy constructor which just copies fields, so the copy and original have the same `Callable_Base` pointer, so when the scope ends, that pointer gets deleted, and our original object is corrupt. God. I love memory management… You're not even reading this, right? Maybe you skipped to the next N things article, or maybe you died of natural causes reading this or something, I don't know. Here, take my social insurance number: **147 504 278**. I even bolded it. Who cares, right? Anyway, we should write a copy constructor that works. Easy. We'd want to initialize `c` with **`new`**… uh. Hmm. **`new`** what? We don't know the underlying derived type, we just know the base class pointer type. We knew the type in the original constructor, but that information has since been lost. This indicates that we should probably again leverage dynamic polymorphism to take care of this for us. Let's see:

```
// ...
template<typename Ret, typename... Params>
struct Callable_Base<Ret(Params...)> {
  // ... our other stuff
  virtual Callable_Base* heap_clone() = 0;
};
// ...
template<typename F, typename Ret, typename... Params>
struct Callable<F,Ret(Params...)> {
  // ... our other stuff
  Callable_Base<Ret(Params...)>* heap_clone() override {
    return new Callable{*this};
  }
```

```
};
// ...
template<typename Ret, typename... Params>
struct Function<Ret(Params...)> {
  // ...
  Function(const Function& other) : callable{other.callable
? (other.callable)->heap_clone() : nullptr} {}
  Function(Function&& other) : callable{other.callable} {
    other.callable = nullptr;
  }
};
```

There. I threw in a move constructor too, just for fun. Now we can actually copy and move things without ruining everything. We've created a pretty good abstraction for lambdas, and it actually works for regular old functions out of the box too! Wow!

After all that hard work, one caveat remains: explicitly writing out the function return and parameter types in `Function`'s template arguments kind of sucks? Like, it would be really cool if our `Function` could just figure it out based on the lambda/function we gave it.

Okay, let's start with regular old functions; they're easier. It's not actually a lot of extra work. The code we've written is such that we could actually do this in a straight-forward way only using *deduction guides* (since C++17). There's a good chance you haven't heard of these before, and that's fine. Basically, a deduction guide tells the compiler how to deduce template parameter types based on the parameters an instance gets initialized with. For example, let's say I have a template class `Thing` that takes a parameter `T`, and if the constructor receives a **`const char*`** parameter, I actually want to deduce `T` as `std::string` instead, and initialize it with that **`const char*`**. Here's how we'd do that:

```
Thing(const char*) -> Thing<std::string>;
```

That's it! Just two lines. "Deduce this to be that". We can also template our deduction guides, and have the deduced parameter shuffle things around. In fact, here's what we'll do for our function deduction guide:

```
template<typename R, typename... Ps>
Function(R(*)(Ps...)) -> Function<R(Ps...)>;
```

This'll take any function pointer the constructor gets, and based on that, deduce the actual parameter given to be `R(Ps...)`, which we have a template specialization for.

Now, what about lambdas? Hahah. So we're actually going to cast our net a little wider than we might've intended in how we do this, because the end result is that `Function` will be able to wrap *any* sort of class that has an **`operator()`** defined, i.e., any functor. Honestly, I don't mind that. Good. Let people wrap arbitrary functors. Lambdas are just functors anyway. Right. So, how can we deduce the return type and parameters in the lambda? It might seem impossible since we don't know what the type of the lambda is, nor whether that type encodes any information about the signature. But you know what does? Its **`operator()`** method. I wasn't kidding: they really are functors, and they have an **`operator()`** method. So, why don't we just check for its return type and parameters? So, let's say we do something like this:

```
template<typename F>
Function(F) -> Function<typename GetSig<decltype(&F::operator(
))>::type>;
```

Here, `GetSig` is some template class (metafunction?) we'll use to turn that **`operator()`** signature into a useful type. Now, you might be asking, why can't we just pass **`decltype(&F::operator())`** or **`decltype(F::operator())`** as the parameter for `Function`? Shouldn't it already have the right type?

Aha. How naive. Another rabbit hole wills itself beneath your feet.

Sit down with me for a moment. You're probably already sitting down, but like, sit down *even more*. Let's talk about a special kind of type in C++. Here's the deal: I'm going to write some valid C++ code, and you're going to stare at it for a while and puzzle over it. Maybe you can hypothesize about what it means. Good?

```
struct Foo { int x; void fun(int); };
// ... in main now,
int Foo::* p = &Foo::x;
```

"What the hell is going on?", is what I asked when I first saw this. That whole last line looks like a typo, but I promise you that it's not. That's real, raw, unfiltered C++. What's it mean, then? Ah. So, in this example, `p` has what's called a *pointer-to-member* type. It's not a pointer to a member of a *particular object*, it's just a pointer to an `int` member of the class as a whole. And when we initialize it to `&Foo::x`, we tell it to be a pointer to `Foo`'s `x` member, generally. That is, for a particular `Foo f`, we have that `f.*p` is equivalent to `f.x`.

Yes, that's how it works. I give you permission to cry. How could you not, in the face of this raw beauty? Here, it works for function members too.

```
struct Foo { int x; void fun(int); };
// ... in main now,
void (Foo::* q)(int) = &Foo::fun;
```

Right. Perhaps it's easier to impart upon you the sense of this with this kind of example. Think about it. I declared **`void fun(int)`** in `Foo`, and if I wanted to define it outside of the class, I'd do this:

```
void Foo::fun(int) { /* ... */ }
```

We can think of that `Foo::` bit as actually being part of the type. And that makes complete sense — what's the key difference between the type of `Foo::fun` and some non-method function of type **`void(int)`**? That's right:

`Foo::fun` takes a first secret implicit `this` parameter of type `Foo*`. That information isn't captured by **`void(int)`**; that's why we need that `Foo::` out front to implicitly communicate that fact. Now, with that line of reasoning, it only makes sense to intuit that a pointer to this kind of method in `Foo` should have type **`void`**`(Foo::*)fun(int)`. And you can generalize this sort of reasoning to the first example with the data member. So then, for a `Foo f`, something like `(f.*q)(42)` would be equivalent to `f.fun(42)`. Good? No? Fine. To each their own, even though you're so hilariously wrong.

As an aside, you might be left wondering whether we can have *reference-to-member* types. I'll satiate your curiosity: the answer is no. Why? Because Stroustrup said so. That's it, that's the reason. Stroustrup didn't really think it was worthwhile, since he couldn't think of any actual use case for something like that. And I mean, if you ask me, fair enough. If we managed to go this long without knowing about pointer-to-member, we probably don't need reference-to-member that much.

Anyway. Back to the point. What was it? Right, `Function` deduction guides for lambdas. Say `F` has a method as follows:

```
Ret F::operator()(Params...);
```

Then, `&F::operator()` will have type `Ret(F::*)(Params...)`. With this knowledge, let's write our `GetSig`:

```
template<typename> struct GetSig;

template<typename F, typename Ret, typename... Params>
struct GetSig<Ret(F::*)(Params...)> {
  using type = Ret(Params...);
};

template<typename F, typename Ret, typename... Params>
struct GetSig<Ret(F::*)(Params...) const> {
  using type = Ret(Params...);
};
```

Lovely. It only took like half a page(?) to motivate that. Note that I included a specialization for `const`, because sometimes the compiler will do that and make a functor's **`operator()`** method **`const`** if it doesn't change any of its member data fields. There might be an edge case or two I'm forgetting, but this is good enough; I'm happy with it, and you should be too! Why? We're done.

```
struct Functor {
  double operator()(double d) { return 42.0; }
};

void foo(float f, int i) { /* ... */ }

int x = 3;
auto fun = [=](int p) -> int { return x + p; };
Function F{fun}; F(8);
Function<int(int)> G{nullptr}; // G() would throw
Function H = Functor{}; H(6.9);
Function K = foo; K();
```

This all just works. It's so nice, right??? We finally have a uniform interface for wrapping all sorts of callable things, and we can pass them around to other functions and treat them as what they are: easy-to-use first class functions. I mean, one caveat is that we can't deduce the template arguments for `G` because **`nullptr`** doesn't exactly give any information about what kind of function this will eventually hold, but that's fine. I leave it as an exercise to the reader to write assignment operators for `Function`. It's not hard, it just wasn't important enough to detail here.

This seems pretty useful! Really, this sort of thing ought to just with C++, eh? Now, with all that out of the way, you can safely throw away all the code we just wrote, because the C++ people have already done it and have probably done it better than us. That's right. `std::function` from `<functional>`. This is literally already part of the standard since C++11, with the deduction guides added in C++17. We just re-made that from scratch.

C++ best language. Anybody who tells you otherwise is a **mathNEWS** editor trying to get me to shut the fuck up.

jeff

O-TEAM SEZ #AD

Waterloo Orientation has opened its frontline leader applications for O-week 2022. Everything is planned to be in-person. Good people, good food, and good leadership experience abound. Apply here: https://uwaterloo.ca1.qualtrics.com/jfe/form/SV_eepXWu5D1uLEZ26

For any questions or concerns, email mathorientation@uwaterloo.ca.

Math O-Team

CUCUMBERS

Jesus could walk on water. I can walk on a cucumber. A cucumber is 99% water. That means I'm 99% Jesus.

CONNOR, MY FRIEND FROM SECOND GRADE

It was late August and Steve stood upon the precipice of greatness, on the dock above a vast lake of cucumbers. The county had drained the reservoir over a year ago for the stunt, which outlets around the world had called the "Greatest Endeavor of the Modern Era". The hills around the reservoir buzzed with journalists hailing from the world over. The air was thick with helicopters recording the event, both those of the press and those of governmental organizations. The coast of the lake glimmered and sparkled with a thousand flashes from a thousand cameras every second. The airwaves worldwide were filled with live streams of the event for seven billion pairs of eyes. Everything was on pause as it waited for Steve to complete what he had set out to accomplish.

Over the past two years, the international cucumber production chain had been rerouted to fill in this one reservoir. New shipping routes had been sailed with cargo ships filled to the brim with containers of millions of cucumbers. New rail lines had been laid, crisscrossing the continent, and were run by freight trains kilometres long with hundreds of specially designed refrigeration cars filled with cucumbers. Tens of thousands of 18 wheelers drove across every kilometre of the country, taking the containers of cucumbers from port to train station, and from train station to reservoir. On the board of the lake, an entire industry of massive bucket wheel excavators and dump trucks had spent the past 10 months dumping the cucumbers into the barren reservoir, turning it from a desert to an oasis of cucumbers kilometres wide.

Steve stood upon the dock and soaked all this in. When he had first proposed the idea ten years ago, no one had taken him seriously. He had been laughed at when people had even bothered to listen to him. Slowly, however, he had found sympathetic ears. Over months and months, those who agreed with him found others who agreed with them, and so on and so forth. Support for Steve began to crop up in many places. People wanted to see what would happen, and the curiosity of the masses is a powerful force indeed. Eventually, word spread to those in high places. Steve had an entire movement behind him, dedicated to seeing his mission through to completion. Dedicated to finding the line between human and god. With enough persistence, powerful people in powerful places began to see his mission as a priority. Money, fame, and power moved mountains, as well as a trillion cucumbers. The world's supply chains were rerouted to this backwater reservoir in the high desert of Colorado. Vast amounts of rainforest land were cleared to make way for thousands of brand-new state-of-the-art cucumber plantations. An army of millions of researchers worldwide made it their mission to find the most efficient way to grow cucumbers, for ways to grow cucumbers in inhospitable climates, and how to squeeze as many plants as possible onto as little land as possible. People all over the world began to grow cucumbers in their community gardens, in their planters, and in whatever green space they had on their lawns. Thousands of logistics professionals worked around the clock for a year to move all these cucumbers to their destination, and today, Steve stood admiring the billions of man-hours that had been poured into his project.

Steve readied himself on the dock. He had been training for this day since he had first had this idea. His mind was completely clear. He took a deep breath and stepped out over the lake.

The hundred thousand personnel gathered around the reservoir held their breath.

Steve's friends, his family, and his earliest supporters — those who had been there when he was just a joke — held their breath.

The billions of people watching on their computers and TVs at home, the billion watching on their phones at work or in bed during the middle of the night on the other side of the world, and the millions more watching on large outdoor screens set up in public squares all held their breath.

The world held its breath.

And waited.

Steve's foot made contact with the top layer of the cucumbers.

And it kept going.

Steve had put his entire weight behind the step, in full confidence, and now he fell down through the mass of cucumbers like a stone. He cut through them like a knife. No one heard his surprised yelp — he was already a hundred metres down by the time he registered what was happening. He kept on sinking, the cucumbers offering no resistance. The cucumbers had been baking in this reservoir unrefrigerated for months, microbes eating them from the inside, and had lost all structural integrity. The only noise the onlookers heard was the squelch of rotten cucumber, as the mighty mass of cucumbers resettled after having not been disturbed in months.

The early chills of autumn froze the cucumber lake a few weeks later, well before the digging crews could make any real progress down. Over months and years, the cucumbers slowly rotted away, eaten by fungus and microbes from within and by insects from without. The stench of rotten vegetables would take years to truly fade from the surrounding valley.

Steve was never found, presumed eaten by the same microbes that ate his cucumbers, the same microbes that ate his dreams.

It is said to this day, all these years later, that the water from that reservoir still has the slightest tinge of cucumber.

aphf

VUCA GIRL

TO THE TUNE OF "BARBIE GIRL" BY AQUA

Hi enamourED! 🌍
Hi World! ✨
Do you wanna go for a ride? 🌍
Sure World! ✨
Jump in… 🌍

I'm a VUCA girl ✨ / in the VUCA world 🌇
Life is drastic 🐯 / I'm elastic 🪆
Fate is playing coy 😏 / just like another boy
VUCA nation 🎡 / thrive 💝 in this creation 🎉

Check this glamour 🤩 you're enamourED ✨

I'm a VUCA girl ✨ / in the VUCA world 🌇
Volatile 🎭 / Fits my style ✨
Catch me unaware / But I don't even care
Random chances 🎲 / Get me my advances 💰

I'm a blue happy girl 🙂
Unpredictable 🤹 world
Mess me up 😩, make me cry 😭, I'm your dolly 💝
But I always bounce back, see the shine ✨ in the sky 🏙️
Don't need certainty here, when you've got me 😁

Dance with me, nothing more 🕺
And I'll lead you back to shore 😊
Vuuuu~

I'm a VUCA girl ✨ / in the VUCA world 🌇
Ambiguous 🤹 / I can do with 🎡
Quirky 🤩 and unique ✨ / so I just stole your pique
Just resilient 👊 / Don't need brilliance 💝

[x2]
Check this glamour 🤩 you're enamourED ✨
V-U-C-A
Check this glamour 🤩 you're enamourED ✨
VU-CA! VU-CA!

Break my heart 💔 with a word
I grow stronger within 💪
Make the stock market crash 📉
I get tax benefits 💰
Head held high like a star 🤩
Shining glamour 💅 in pink
Hit that beat with my heel 👠
Sing it with me 🎶

Turn around, 🔨 break 🔨 my plans
But I've always got my friends 😁
Turn around, 🏴‍☠️ steal 🏴‍☠️ my grants
I'll still say it to the end

[x2]
Check this glamour 🤩 you're enamourED ✨
V-U-C-A
Check this glamour 🤩 you're enamourED ✨
VU-CA! VU-CA!

I'm a VUCA girl ✨ / in the VUCA world 🌇
Life's erratic 🎲 / still fantastic 💝
World blew up again 👊 / Pandemic conflict pain train
While you're shakin' 😖 / I eat 😋 VUCA daily

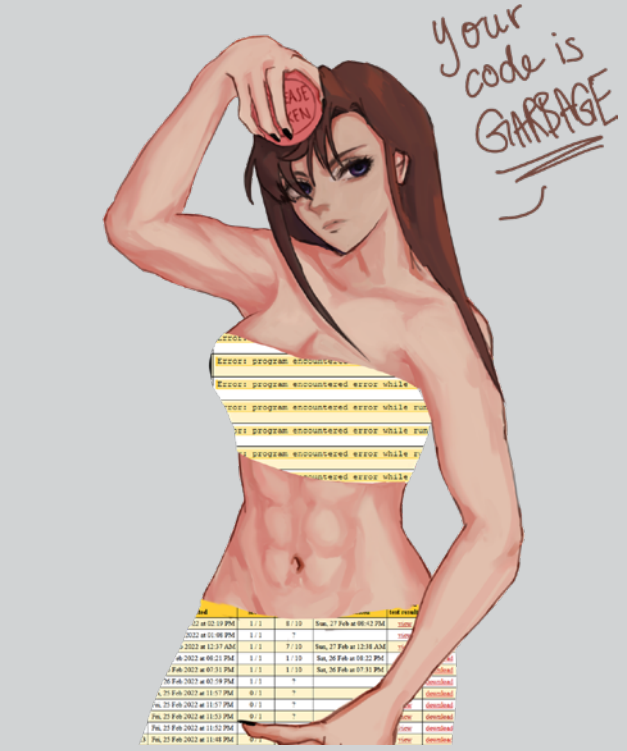
I'm a VUCA girl ✨ / in the VUCA world 🌇
Life is drastic 🐯 / I'm elastic 🪆
Fate is playing coy 😏 / just like another boy
VUCA nation 🎡 / thrive 💝 in this creation 🎉

[x2]
Check this glamour 🤩 you're enamourED ✨
V-U-C-A
Check this glamour 🤩 you're enamourED ✨
VU-CA! VU-CA!

Oh, I'm having so much fun! ✨
Well, enamourED, we're just getting started 🌍
Oh, I love you, World! ✨

[redacted]
(get **mathNEWS++** *to find out this writer's real identity today!)*

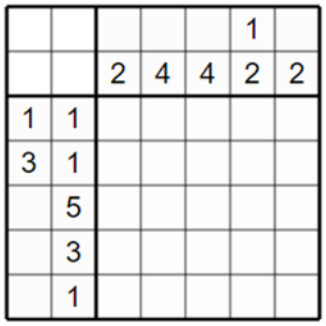
MARMOSET-CHAN



Couch

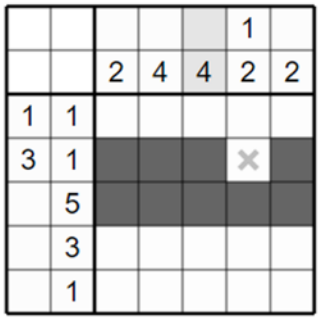
HOW TO SOLVE A PICROSS

Picross are picture logic puzzles in which cells in a grid must either be filled or left empty based on their corresponding row and column labels. Here's an example of what a picross puzzle might look like:

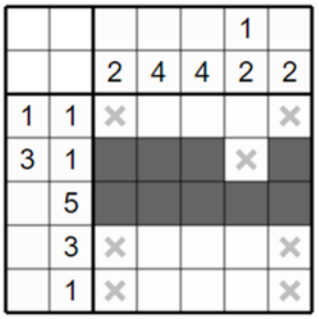


Each row and column is labelled by a series of numbers. These numbers represent the lengths of blocks of repeated cells, in order, where each block is separated by one or more empty cells. For example, row 2 means that there is a block of 3 filled cells, followed by 1 or more empty cells, followed by a block of 1 filled cell (going from left to right). Another example is that column 4 means that there is a block of 1 filled cell, followed by 1 or more empty spaces, followed by a block of 2 filled cells.

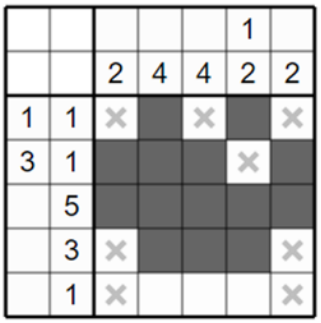
Let's walk through solving this picross. To begin, we can immediately fill out row 3, since there are only 5 cells in total in the row. We can also immediately fill out row 2, since there is only one way to place a block of 3 filled cells, followed by 1 or more empty cells, followed by 1 filled cell. Here's what the puzzle looks like after filling these in:



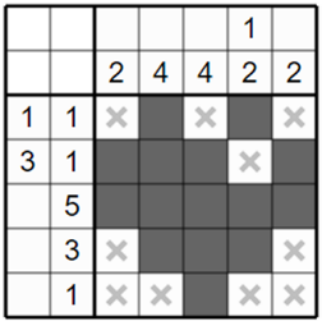
Next, we see that columns 1 and 5 both have all of their blocks, since they each need a single block of 2. This means that all remaining cells in those columns must be empty, which we can mark off as follows:



Now, with the remaining unsolved cells, we see that there is only one way to fill out the first row, since there are only 3 unsolved cells left. We also see that all of the unsolved cells in row 4 must be filled as well. This leaves us with the following:



Finally, we can solve for column 3 by filling in the only remaining cell in that column to meet the requirement of a block of 4 filled cells. This also satisfies the requirement for the only remaining row (row 5), which means that we're done!

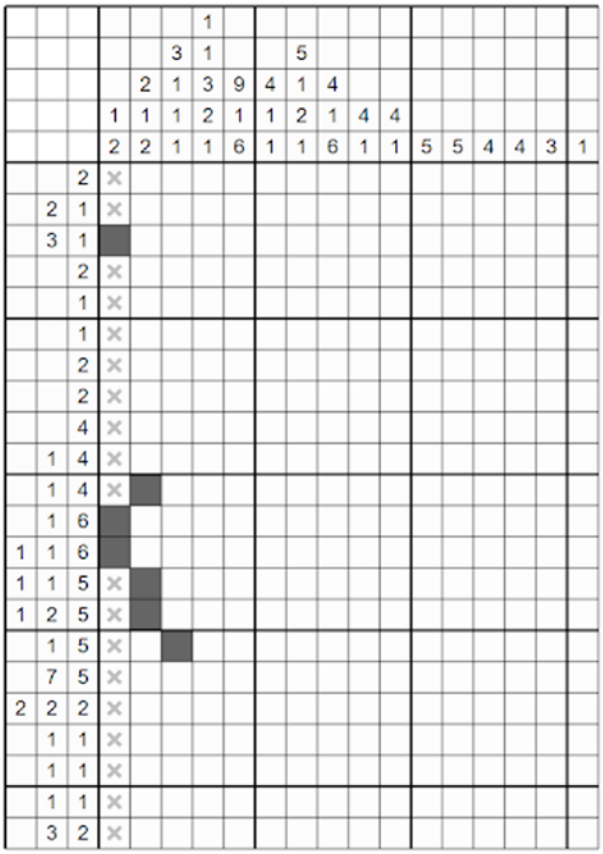


Hopefully that wasn't too painful! If you'd like to try some more picrosses on your own, I recommend checking out https://puzzlemadness.co.uk/picross. Once you're a picross pro, give this one that I made a try. You might see a familiar face!

MISS THE FILLER? SUBSCRIBE TO mathNEWS++ AND GET 30% OFF A LAZEEZ CHICKEN ON THE ROCKS!

ONLY ~~30~~ 25 FLITCHES OF BACON PER BILLABLE FLIGHT HOUR

The first column, as well as several other cells have already been marked for you:



John Writer

[Editor's Note: If you're a **mathNEWS** veteran, you might also recognize these by their more **mathNEWS**-y name, nonagrams.]

CATCH!

Hey, just want to say, catch!
Catch the snowball coming at you.
Catch your dreams before they fade.
Catch the bus before running late.
Catch up with the sleep you missed.
Catch the moment and do whatever is right at the minute, and don't look back.
Catch the cockroach under your bed. It's disgusting.
Catch the flag of the opposite team.
Catch the present time because it'll be flying and never coming back.
Catch the love you can get and catch the things you need.
Catch the stories of people around you. Some of them are great. Other are rubbish, but hey, it might still be a great quality time.
Catch the knowledge in class. Could be useful.
Catch. There's so much to do. So much to explore. So much to feel. So much to listen to. So much to catch.

Catcher

UW WRAPPED

If you sign into LEARN after all your courses are over, half an hour (give or take ~10 seconds) before the start of the next term, you'll be graced with a little-known Easter egg. It's called *UW Wrapped*, and it is glorious.

It's been there since at least Fall 2019, when I first ran into it by accident. I've made a point of seeking it out every term since to figure out what's going on. The theory I settled on is that *UW Wrapped* is the secret passion project of a UW co-op student at D2L, the company that made LEARN. So, I kept it a secret for years in case the student could get in trouble if the wrong people heard about it. But heck, this unsung hero probably graduated by now, so it should be safe to reveal.

If you've seen *Spotify Wrapped*, you know where this is going. After you log in, a vertical modal appears in the middle of the screen. It begins displaying a bunch of highlights from your term (though unlike Spotify, everything is in LEARN's classic black and white theme). As a teaser, here are some of their titles.

- **Top Classes**
- **Biggest L**
- **Bragging Rights**
- **Secret Admirer**
- **Secret Admiree**
- **Cheat Sheets** (this one makes me pretty paranoid)
- **Best Friend**
- **Degen Hours**
- **Last Minutes**
- **Biggest Clutch**

You can probably guess a lot of these, but if you're curious what some of them mean, then check out *UW Wrapped* for yourself at the end of this term.

I unironically love *UW Wrapped*. University life is often so blind to itself. In the short run, it's reactive: see deadline, do deadline. In the long run, it's ambitious: get grades, get jobs. *UW Wrapped* shows you how your term actually was: not just the next week of deadlines to panic over, not just the grade at the end. These were your highs, your lows, your last minute grinds, and the people who were along for the ride with you.

Of course, *UW Wrapped* only sees what LEARN sees, so it's not a perfect view of your term either. It's a bummer that our digital experience at UW is fragmented across so many places. How cool would it be if we could also have our Piazza stats and Crowdmark highlights too? WatCard purchases, buildings we are in? The information is out there somewhere, we just need a hero to tap into it.

If you're reading this, dear hero, thank you.

water

AWESOME POINTS V

CONTINUED FROM LAST ISSUE'S AWESOME POINTS

The fifth floor room of MC was filled with the sound of chatter. The emergency council meeting had arrived, and Name was ready to shut down Gaming Club's Awesome Points program.

Maria, the MathSoc council speaker, called the emergency meeting to order with rap on the desk.

"Good evening, councillors. As I'm sure you've all read in the agenda, we're meeting today because of a motion put forth by councillor Person. I'll let her speak to it now."

"Thank you, Ms. Speaker," Name began. "Whereas Gaming Club has launched Awesome Points, an incentive program which irresponsibly uses funding to give monetary rewards to participants in online chatrooms—

"And Whereas Gaming Club has extended the program to members of the public, thereby misusing MathSoc club resources—

"And Whereas Gaming Club has violated MathSoc communications policy by running advertisements for Awesome Points on various social media platforms—

"And Whereas Gaming Club has ignored concerns about the Awesome Points program brought forth by councillors, demonstrating outright recalcitrance in their responses, which include bribing MathSoc councillors with gift cards—

"Be it resolved that Gaming Club terminates their Awesome Points program, that Gaming Club is put on probation and sanctioned for any MathSoc funding for the next year, and that Gaming Club's executives be removed immediately."

The ensuing debate was ferocious. A good number of councillors were apparently enthralled by the popularity and fame of Awesome Points. Maria could see that there would be no changing anyone's minds, and after half an hour, the question was split and called.

Maria held the vote, MathSoc terminated Awesome Points and placed Gaming Club on probation, 18 to 16. Fury mixed with triumph in the silence after Maria announced the decision. Arthur looked like he was trying not to cry.

The fate of the Gaming Club's executive was a more mixed decision, and the vote was tied, with Maria holding the tiebreaker vote as speaker.

Hope you've learned your lesson, thought Maria, and she decided to spare Josephine.

❦

Arthur returned to the Gaming Club meeting room exhausted and defeated.

"It's over. MathSoc is shutting down AP, and we're on probation with sanctions now. But… at least we haven't been removed from the exec team."

Josephine took a deep breath in and held it. Only after Arthur started shuffling did she let it go. "It's okay, Arthur." She turned. "Well, President lightSoul, I suppose Operation Jupiter is going into effect after all. What's the latest offer from ManaCorp?"

The cloaked president slid a slip of paper to Josephine.

A melancholy smile broke across her face. "Every cloud has a silver lining, I suppose. That can keep us running for years. Who would have thought Awesome Points would be worth that much? It's been a good run, everyone. Meeting adjourned."

lightSoul grunted in triumph.

❦

"So, we killed Awesome Points," Blas said, back at the SLC lunch table the next day. Are you happy?"

"It brings me absolutely no pleasure in bringing punitive measures upon Gaming Club," Name said with a ferocious smile, "I hope this teaches them a lesson in responsibility."

"Did you know they sold Awesome Points to ManaCorp?"

Name, who was halfway through a bite of salad, almost choked. "What?! They can't do that!"

"They are a magnificent club." Blas shrugged and shook his head, still grinning, as Name jumped up and ran off with that same determination Blas had seen so many times before. He raised a can of soda. "Awesome Points is dead—long live Awesome Points!"

CC

N THINGS TO DO INSTEAD OF YOUR CS ASSIGNMENT

- Stress over said assignment
- Stress over stressing over said assignment
- Eat free pizza at **mathNEWS**
- Write an N things article about said assignment (infinite recursion)

peacelovemath

profQUOTES 148.6

CO 250: JOCHEN KOENEMANN

❝ To make things simple I will abbreviate it with CS. No, that's not computer science, but rather complementary slackness.

CS 136: TIM BRECHT

❝ *(directly after covering graphs in an assignment)* Graphs are the only core data structure we are not working with in this course.

CS 146: BRAD LUSHMAN

❝ If you wanna call it cheating, call it cheating. But I call it sticky pages.

CS 350: EMIL TSALAPATIS

❝ Assignment 3 is… well, I wrote it, so I think it's easy. But you might disagree.

❝ Modern software is frameworks on top of frameworks on top of frameworks. (…) Turn on Slack, it takes eight gigs of memory and five seconds to load.

❝ The only time I understood a filesystem was when my prof told me, "You have two weeks to build a filesystem, go."

❝ With memory, you reboot and your problem goes away. With filesystems, your problems are for life.

CS 442: GREGOR RICHARDS

❝ Algol is notable for basically inventing structural programming, but more importantly losing the war to C.

❝ I'm personally using the Commodore 64 to stoke my own nostalgia.

❝ Xerox PARC, who were well-known for inventing literally anything and failing to make any money off of it.

❝ The most popular object-oriented programming language right now is probably JavaScript.

❝ Abstraction is useful. Hopefully that is such a vapid statement that nobody disagrees with me.

❝ I realize this is incredibly boring.

❝ If I said, `Object subclass: BetterSquare`… note that this is not actually going to be a better square.

❝ I'm trying to think of the most divisive way to do this.

❝ This is dynamic dispatch, and this is the reason why compiling object-oriented code is misery.

❝ Structural subtyping is, "If it quacks like a duck, it's a duck."

❝ I hate computers and love boards.

❝ Don't ask me what that means — I have no clue what it means for a rectangle to have width "linked list".

❝ `"hello" + 2` is nonsense, unless you're using JavaScript, which will happily change anything to anything.

❝ Let's take this moment to evangelize Debian.

❝ God, I just called Java a purely object-oriented language. How perverse.

❝ In our formal semantics, we would say this "gets stuck".

❝ We're the ones writing the type checker, we can put in whatever stupid rules we want to.

❝ Except-for-null type safety is *almost* type safety I suppose.

❝ Secretly, the object-orientation module is the polymorphism module.

❝ Nobody understands the C++ overloading resolution algorithm. The author of the C++ overloading resolution algorithm certainly does not understand the C++ overloading resolution algorithm.

❝ There's the solution to our initialization problem. Our solution is, "Eh, just throw nil everywhere, what could go wrong?"

❝ I can't put every number in the dictionary, I hear there's an infinite number of them.

❝ Now, if I was a Neanderthal imperative programmer…

❝ God, why would you make [Smalltalk] like this. It's like they never had to use their language.

❝ We're not going to do anything useful with concurrency, we're just going to have it be a problem that we have to solve.

❝ Let's be honest, GPUs are just glorified vector programming machines.

❝ If you're wondering why I'm inconsistent about that — I'm consistent, I'm just a horrible pedant.

❝ I kind of wonder if [pi calculus] was designed as a joke, it is that bad.

❝ `replicate` — written exclamation point because English is for losers…

ECON 371: PREDRAG RAJSIC

❝ You're not stupid, you're just emotional.

MATH 148: BLAKE MADILL

❝ I'm trying to rush because I have -1 minutes.

❝ Is there a non-zero chance this appears on the exam? *[very long silence]* No, there is a 0% chance this shows up on the exam.

❝ It's good enough if you're an engineer and building an e^{-x^2} shaped bridge or whatever they do. Who knows. We can never be sure.

❝ I can't have you liking the cat more than me. It's bad for my ego.

MATH 249: KEVIN PURBHOO

❝ This should remind you of a delta-epsilon definition, except there's no deltas or epsilons.

❝ Welcome, everyone, to the real world.

❝ Please don't vomit symbols on the page.

❝ The thing you want to avoid is, you want to avoid being wrong.

❝ I could try to simplify this, but like, yuck.

❝ The idea is: to give you an idea.

❝ Forget everything you learned so far.

❝ *[Points to a graph]* Nobody likes this.

❝ If you talk about the empty graph with mathematicians, you will likely start an argument.

❝ Is there an algorithm? Yes! Is it efficient? Hell no!

❝ The reason this is not a big deal is because it's not a big deal.

❝ I don't know what I just said there.

❝ We could move on, but it's Friday, so I don't wanna.

❝ I claim that you all knew this.

❝ We're out of time, but I wasn't going to prove this anyway.

❝ I forgot I'm supposed to tell you every day that you have an exam coming up.

❝ What I said before… that might not exactly be completely true.

PHIL 121: MATHIEU DOUCET

❝ We don't eat robots, but we do eat chickens.

❝ Stairs suck!

❝ There was no moustache twirling or thumb twiddling.

❝ I'm not completely clueless; I'm just confused.

❝ I used to be a military tour guide in a previous life.

❝ Everyone always forgets about Manitoba.

❝ There are two universities in Canada that are majority male. One of them trains armed military officers. The other is us.

PHIL 145: VANESSA CORREIA

❝ Any economists in this classroom? *(Nobody says anything.)* Oh, thank God.

❝ I am very solidly a round-earther.

❝ If you want to find the truth, take philosophy. You'll never find it though. They suckered me in for 9 years before I realized that.

❝ Let's think critically about whether this bus will hit us.

❝ I'll test [gravity] by dropping laptops and cell phones and other really expensive things.

❝ I'll keep doing this until I've broken all your valuables.

❝ Hopefully you get something other than shampoo ads.

❝ I'm sure you've already forgotten most of what you've learned.

❝ If my face swells up and I stab myself in the leg, class is over.

❝ Oh, I don't mean a knife, I mean an EpiPen.

❝ I was having a weak Instagram moment (…), so I clicked the ad.

❝ You probably don't want to be applying headphones directly to the wound.

PMATH 351: BLAKE MADILL

❝ I was here in the nick of time *[points to student named Nick]*.

❝ Valentine's Day on Monday! We'll celebrate with a midterm; I'm here to break hearts.

❝ And now I'm going to shamelessly look at my notes.

❝ The shell of me is here and ready to prove the Baire Category Theorem.

❝ I'm still traumatized by Zoom school.

❝ Let's see if I can do fractions today.

❝ So this is the Cantor set. Who cares!

❝ We finished Week 8 on time! We did it folks.

❝ The proof isn't so bad. Especially if you leave out the details.

❝ I do so much high level math that I have no use for such three digit numbers like 911.

❝ Don't put that in your notes. That's embarrassing. Don't write that down.

❝ This follows from the disjoint assumption from Mr. Hausdorff.

❝ Oh that is way overkill! You sledgehammered the board; we can never use it again.

PMATH 352: SPIRO KARIGIANNIS

❝ Student: Which do you like better, complex analysis or real analysis?
Prof: Geometric.

❝ A more sophisticated way to think of this is to think of it as a Riemann surface, which is a manifold, which is a *[technical mathematical jargon that the student recording this did not understand]*. We're not going to think that way.

❝ As you can see, I have legs.

❝ Sort of completely true what I just said.

❝ This is obviously a faster method if you do it right. Which I didn't.

❝ As you can see I'm not strong at arithmetic.

❝ There's going to be people who will want to strangle me after this midterm.

❝ The bell distribution would only work if I had infinitely many students, so I don't curve.

❝ All letters should have one connected component.

❝ What doesn't kill you, really hurts.

SMF 211: TONI SERAFINI

❝ Now we can see some of the most used emojis for sexting include: eggplant, peach, and, *ahem,* moisture droplets…

STAT 241: MICHAEL WALLACE

❝ If you torture numbers enough, they'll confess to anything.

❝ I promise not every example is about COVID, it's just… *infinite data*…

❝ If I enter more than 5000 data points I'll get in trouble with the Math Faculty computing centre.

❝ This is obviously completely ludicrous.

❝ We're going to call these drugs A, B, C, D — for Awesome, Bodacious, Cool and Delicious.

❝ All hail Central Limit Theorem.

❝ Why did I put a square there? Naughty!

ENJOYING THE profQUOTES?

SUBSCRIBE TO mathNEWS++ TO GET EXCLUSIVE ACCESS TO EVEN MORE OF YOUR FAVOURITE QUOTES!

USE PROMO CODE STAT231 FOR A P=0.01 CHANCE TO GET 30% OFF YOUR FIRST N MONTHS!

ARE YOU THE ONE PART 2: QUESTIONS AND PROBLEMS!

(Special thanks to Imax, who was a huge help with the math for this article!)

Hi **mathNEWS** readers! I wanted to revisit a topic that I wrote about in my previous article, being a reality TV show called *Are You The One* (available on CTV streaming for free, but please use an adblocker for your own sanity). In case you didn't read the first article, I'll give you enough info for 2 related problem specifications:

Are You The One: the straight (bipartite) version

- n (straight) men and n (straight) women (contestants) are given psychological profiles to find a "perfect match" and sent to an island.
- The contestants' initial information is that their perfect match resides on the island with them, but not who.
- If all contestants find their perfect match in an allotted m week timeframe, they collectively win $1 million. Otherwise, they lose.
- Every week, the contestants gain information in two ways: firstly, one pair may be elected to enter a "truth booth" (TB) in which the pairing is confirmed or denied, and there is a "matchmaking ceremony" (MC) where the $2n$ contestants form n distinct pairs (monogamous, heterosexual), and are revealed the number of correct matches, but not who.
- Getting no new information in an MC (e.g. 0 correct couples, 1 correct couple with 1 confirmed TB), then their prize money is cut substantially (but not lost). This is known as a Blackout.

Are You The One: the bisexual (complete) version.

- This time, $2n$ contestants form n couples, but any among the $2n$ contestants can pair with any other.
- Everything else is the same as the heterosexual case.

With this in mind, it's natural to ask a few questions that may apply to both cases:

1. What are the number of possible pairings?
2. At the start, what would be a strategy that avoids Blackout?
3. How do the contestants maximise the information they gain in each week/is there a way to "brute force" a win?

The first question is straightforward, relatively speaking. In the bipartite case, we have $n!$ combinations as one subset "chooses" distinct edges to the other. In the complete case, with a little more legwork, we can find something that's still rather intuitive: In the case of $2n = 2$, we only can make one possible pair. With $2n = 4$, person 1 can either pair with person 2, 3, or 4, which automatically "sets up" the people left out together. So we get $(1)(3)$. This is like an "odd factorial" which makes sense since 2 people are removed at once to make a pair. This can be written as $(2n - 1)!!$ (the double factorial meaning only multiplying numbers with the same parity).

Where the first question is one of combinatorics, this one is one of algorithms. Let's first think about random assortment of couples, how good is that? Well, there's this famous similar problem that goes "if a butler is guessing to give people's hats back randomly, what's the probability he messes all the hats up?" Sparing you the details, the probability of a derangement occurring approaches $\frac{1}{e}$ as n approaches ∞, which is about a 37% chance. Could one try to achieve better odds? Here's one possibility: the contestants give equal time to meaningfully talk and get to know all of their suitors (also contestants). Then, each contestant gives a distinct rank to each of their suitors. In the bipartite case, a stable marriage algorithm can be applied from here. In the complete case, stable marriage doesn't directly apply (because stable marriage requires a bipartite graph with positive distinct edge weights), but one could employ a similar algorithm:

- Contestant A_0 chooses their highest rated contestant, who "maybe"s their proposal, all the way down to contestant A_{2n} in the complete K_{2n} graph
- Any contestant may "trade up" their proposal if they receive another that they prefer.

I've tested this on small graphs and it "seems to work" but I haven't actively proved this algorithm's correctness. I'll let you guys know if I get around to it.

Last question: can we brute force this? Now, I barely know any information theory, so Imax was a huge help with this. Firstly, let's focus on the easier bipartite case, where n represents both the number of each subset, as well as the number of couples required. We can apparently trivially upper bound the amount of information we need as n^2, since that's just multiplying the cardinality of the subsets together. After a lot of work, he determined the lower bound on the information required to be $O(n^2)$ as well, with an argument of trying to recursively reduce the size of our subsets by taking out pairs, and getting something similar to $\frac{n(n+1)}{2}$. From there, he calculated the maximal amount of information one could get in a single week, being $O(n)$ by checking edge (i.e. potential match) permutations. Over m weeks, that leads to $O(mn)$ maximal information. So, if m is sufficiently large (i.e. matches or exceeds n), a brute force of the game is possible. Of course, information gained each week need not be maximal, so ideally an m even bigger than n is better for brute force. Part of the game, however, is "playing with one's heart," which may hopefully allow contestants to reduce the problem size by reducing who they're considering as suitors. We didn't end up getting to properly evaluate the complete case (let's be real, it was mostly Imax teaching me about this stuff), but Imax did mention that the complete case would probably be similar and not too more difficult to step through.

Does all this apply to real life? Well, to some extent. Most contestants have an understanding of basic strategy, like keeping one's options open. But, as is common in reality TV environments, the contestants are pressured to drink, provided with no writing implements or technology access, are constantly in close proximity with little room for personal space, privacy, or planning, and forced to do weird challenges and be in strange situations. In short, it's a psychologically taxing experience, and I doubt any of them are going to memorize stable marriage and read an information theory textbook before going. So take all this as a fun exercise that emerges from a real life situation, rather than an invitation to try to be casted on *Are You The One* (though by all means, go ahead, and also send **mathNEWS** your audition tape)!

I hope this article inspires people to look for the math in their every day lives, as all of this came from a mediocre MTV reality TV show, where I was just watching because the contestants are hot and dramatic.

Xx_420SonicFan69_xX

STATS BAD XD

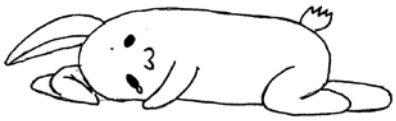
Earlier today, I was having some grass-is-greener syndrome with UofT. It began when I was looking into our school's statistics department.

Students online complained about the poor introduction to the subject — STAT 231 — and expressed abhorrence to the course (long R assignments worth very little that don't help with the exams, most of the actual stats showing up in the last 3 weeks, other administrative issues[1]). Upper-year students seem to be facing a low ceiling of success, at least compared to students in Pure Math and Combinatorics & Optimization who are commonly seen to be taking grad courses in terms 3B and later. Meanwhile, statistics students need a 90% CAV minimum to take 3xx courses before 3A, a 90% CAV and 4A status to take a grad school course. Hearing some other course-organizational and administrative complaints, I remembered that I had an offer from UofT for their statistics department. What I found was quite funny was that students from UofT, on their subreddit, had a hilariously similar level of enmity towards their statistics department.

Is there some universal force of statistics departments being annoying everywhere? I wouldn't be able to get any causal evidence because, like many of the students I talked to, I didn't pay attention in stats class.

Sahil

1. Source: Wink Wonk



MIMETIC RIVALRY: FAKE COMPETITIONS AND UNLIMITED DESPAIR

It's often said that human beings are creatures of imitation. We think our wants spring from the heart. More often, they jump from person to person, like a mild case of omicron. Children learn about the world by imitating others' basic actions. Adults imitate others' desires. This is how the late Stanford professor René Girard saw the world.

When bringing up social media, fashion, or three hundred people gathering to watch an anonymous redditor streak through SLC, most people might be inclined to agree with him. Even when thinking about other subjects, we can totally see people getting degrees because their friends are getting degrees, or because some teacher told them it's a good idea, even if they couldn't care less about their field. What about math? Most of us think the field is unique, and we definitely don't go to university for the silly reasons that every other subject does. That's true in part, but many might relate to this story about Goose420, a hypothetical math student based on a few of my friends.

Goose was pretty generally talented throughout high school and applied to Waterloo for the name recognition. The most desired programs were math and CS. He knew he could get in, so he did. At Waterloo, Mr. 420 met some new friends. Most were around his level, some were way better. He tended to gravitate towards the better ones, joining the same clubs, picking up the same habits and going to the same events. His half-finished side projects always seem to be whatever hit topic is in the news, whether it's NFTs, biotech, retail trading, or the most recent machine learning paper.

If this isn't you, this is probably someone you know. If it is you, Girard can tell why you probably feel sad and exhausted. He coined the term *mimetic rivalry*, describing when people copy each other's desires and compete to get the most of that desire. It can happen with degrees, hobbies, food, and even friendships. In the end, too many people are competing and only a few can win. The only winning move is not to play.

Ask yourself how much you care about the things you put your effort into. Would you still care if nobody else did? Is any of it good for yourself and your future? And how does the *mimetic rivalry* of others affect you? Make sure your desires are shaped by you, rather than by everyone else. The odds are, that's the first step to a happier life.

fluffiest cactus

EVER WONDERED WHAT THIS BUNNY WOULD LOOK LIKE IN COLOUR?

mathNEWS++ • ONLY 7 SHMECKELS PER JIFFY

GAMER BOY: A PARODY OF SK8ER BOI

He was a boy
She was a girl
Can I make it any more obvious?
He was a gamer
She didn't game
What more can I say?
He loved to game
She'd never game
Secretly she wanted to game as well
But all of her friends
Were in denial
They had a problem with his gamer style

He was a gamer boy
She said, "See you, lamer boy"
He wasn't good enough for her
She had a pretty frame
But her head was pretty lame
She needed to play an epic game

Five years from now
She sits at home
Doing boring shit, she's all alone
She goes online
Guess who she finds
Gamer boy being so gamer fine
She calls up her friends
They've seen the sight
They've followed him on all the gamer sites
She starts to watch
And lurks in the chat
Watches the gamer that she scoffed at

He was a gamer boy
She said, "See you, lamer boy"
He wasn't good enough for her
Now he's MLG pro
And he do be gaming doe
Does your normie face see what he's worth?

Sorry, girl, but you missed out
Well, tough luck that gamer's mine now
From you, I am an upgrade
This is how the game is played
Too bad that you couldn't see
See the man that gamer could be
There were easter eggs to find
And I found the pro that is inside

He is a gamer
And I am a gamer
Can I make it any more obvious?
We are in love
Haven't you seen
How we play on the same team

I'm with the gamer boy
And he is no lamer boy
I'll be right there for every life
We got a gamer mic
We're playing a game we like
And we say, "Yahoo! Let's-a-go!"

someBODY

A SHITTY POEM TO THE MATH FACULTY

Dear Math Faculty,

When I first arrived I was seventeen and little did I know you would be so mean. You made me lose all my self esteem, for many nights in MC I tried to not be seen.

WDs made me cry, I knew I could kiss the credits goodbye. My phone struggled with Eduroam Wifi and I spent many nights wishing I was high.

In first year I didn't sleep for four days straight and I was wondering if insomnia would be my final fate. Thank god for my amazing roommate who hated integrals and could totally relate.

I'm guessing you think this experience was depressing, but underneath all the stressing it was actually a blessing.

I was excited when I applied and it's been one hell of a ride! 5 years later and my pink tie still fills me with pride, it's my favourite memento that I'll never hide.

Naps in the MC comfy lounge kept me sane, despite weekly assignments that were a drain. I dreamt about how I'd rather be in Spain, but knew with this degree those dreams I could attain.

I'll always remember waiting in the infinite Pi day line, where the selection was divine. The wait was completely fine because I knew that delicious pie would be mine.

I wrote for **mathNEWS** on Monday because Pizza is the best pay. And though I may go away the memories always stay.

UWaterloo math professors made some truly memorable jokes and the facts they spewed were no hoax. From my brain a lot of knowledge they did coax. What an outstanding bunch of folks!

So now these final words for my fellow math nerds: while the line between love and hate was definitely blurred, an unforgettable time occurred.

Two soon to be math grads

BEING A HIPSTER

TITLE LEAVE YOU HANGING?

SUBSCRIBE TO mathNEWS++ TO VIEW EXCLUSIVE ARTICLES LIKE THIS AND MORE!

FOR THE LOW LOW PRICE OF YOUR FIRSTBORN CHILD EVERY MONTH

I LOVE MONEYMAKING!

Official Merch I would love t-shirts of some of **mathNEWS** *cover or facemask that would scare away covidiots with math*

BEYOND META

Lower prices to an extreme degree for a couple of years, and burn through investor capital doing so, but keep reaping more investments from VCs and the public through the unsustainable growth so far experienced, leading to higher and higher valuations, eventually capturing the entire market and driving all competitors out of business due to said low prices, effectively creating a monopoly.

Then we hike prices

APHF

Collect funds for the construction of MN, a new Math building on campus, where N is a number arbitrarily large. The funds will be used to construct the new building once all Mn buildings, where n < N, have completed construction.

BOLDBLAZER

mathNEWS *charity marathon (all proceeds go to sponsoring a random UW student's education). We record it and film their reaction as they see their student debt being erased. Then they have to pay taxes on it next April and we film our "Epic Prank" video then. I lied, this is not a marathon, this is a YouTube channel.*

TENDSTOFORTYTWO

[redacted]
(get **mathNEWS**++ *to find out this writer's real identity today!)*

WHO DOES THE WEATHER WORK FOR?

With the recent developments in weather, it is clear that whatever function is representative of the temperature is not continuous, or looks like one of the worst functions ever written. Someone please hack life already to fix this, I will reward you with a free subscription to **mathNEWS**++ (the greatest commodity anyone could ever ask for, source: dude just trust me).

I feel sorry for the guy who was running around SLC with a mask covering his juicer.

Wink wonk

WHY ARE TWINS?

Disclaimer: I harbour no ill will towards any real-life twins who may read this. I would not like to fight both of you at once.

For a brief thought experiment, spend a minute and try to think of famous twins not currently alive. Like me, you will probably blank, or resort to mythological examples. Religion is littered with examples of twins, from Artemis and Apollo to the creation myth of almost any dualistic faith. History? Not so much. Are twins destined to fail? It may be possible that there is a psychological factor involved with the Gemini that prevents its members from achieving greatness. The fame of most modern twins seems largely intertwined with their very status as such, being useful as identical actors, or doubles partners in tennis. In the scientific arts, on the other hand, there seem to be few fitting examples (RIP to the Bogdanoffs). One can look to nurture, perhaps at the inability of a parent to instill the proper individualism the same in two than one. This sense of a unique self may be of major mental assistance towards ones' drive to distinguish oneself from others. Jumping to conclusions, we see that when one is so dragged down by their biological inferiority and mental equivalence to another, they find it much more difficult to obtain prominence. We will now leave these unprovable psychological assertations in the past and move on to why twins are funny.

We are obsessed with the concept of twins, with the beautiful idea of the two that is not and never was one. There is something deeply unnatural about the twin, a perfect duplication so rarely seen in nature that has so often ended in twin infanticide. Gemini is mathematical, a multiplication by two, a perfect translation echoed in the world full of slight imperfections and deviation such implied by the nature of DNA. Look at Tweedle Dee and Tweedle Dum of *Through the Looking Glass*, a mathematically involved story that uses its twins as the peak of its humor. The Tweedle twins exhibit humor directly through their mathematical duplication yet ever so slight dissimilarity, in the way a single character never could. In *The Matrix Reloaded*, The Twins as programs in the Matrix signify the artificiality of the Gemini as a concept and its association with the inhuman. At the same time, they are humorous through their very being of twinhood. The same is echoed in the Weasley twins, Patty and Selma in *The Simpsons*, and the twin bouncers in the movie *The Batman*. But what about these properties of twinhood implies humor? The very essence of humor is an active philosophical debate, but several big names including Kant, Schopenhauer, and Hegel postulated humor to derive from unexpected incongruity. Our image of the world expects one reality, but that expectation is "disappointed" by an incongruity. What is more incongruous than such an unnatural, mathematical concept appearing perfectly reflected in the natural human? This same humor is unfortunately reflected in real-life identical twins, as anyone who has met a pair may be able to attest.

It begins to seem a rather miserable fate to be born a twin. By all likelihoods, it may be. The mere presence of a sibling is harmful to ones' future economic success, according to studies I am misconstruing the results of. To be thrust into an almost competitive situation with your twin, and to be perceived as humorous through this relation, may be antithetical to personal accomplishment in life. There are obviously much more disadvantaged situations to be born into, but how many others exhibit such comedy?

Turing Machine

THE GOOSE DEALER RAP

I once saw a meme that the birds at the park are free
So it was the sign to begin my full stealing spree
And as you can see, the police—they ain't unto me
This ain't just a feat, they are beat, it's a robbery
Now my hunger just begun, saw the devil 'fore my eyes
When I roll up to prod night, I don't need to don disguise
Cause my crime is rising high, why should i anonymize
Catch me hustle waterside, all I do is terrorize

All the city now knows there's a new thief on the loose
So I best believe you got it if you wanna buy a goose
If you come in with a wire you will get caught in a noose
Cause your chirping made me tired so I got a flock to soothe
And despite all of you hating I am still here on my ground
Living on in infamy while you're waiting to get your round
Aviary sounds ring loudly but hush money blocks the sound
Last duck dealer to cross me
Floats near the river's lost and found

[Chorus]
A goose dealer, purse filler
Those at the zoo only know me as a truce killer
flip the bird so I won't ever lift up two fingers
My opps said they wouldn't talk but now they're truth singers

Big bird couldn't stop the trade war from going on
There are families on their walks when I fight over my pond
Golden eggs aren't a myth, mother goose has made us strong
Now over half the blocks just couldn't wait for my swan song
So I plan to become rich and only move to prayers then
All bird dealers go to hell so why waste time and play pretend
I make mills so when I pass I'll carry more when I descend
You are here with 0 chicks and still ask why we aren't friends

Any goose of yours gets seized, under any tree or bush
And my pocket's full of green, but it isn't cash or kush
Yea I gotta store bird feed, otherwise in a quick woosh
The geese's health will go downhill
And my bank account gets mushed
So come find me if you wanna make yourself rich with geese
If you know that my goose down beats your 20 dollar fleece
But if the smell of geese food will disturb your rest & peace
The only fowl you'll ever get
Is the foul smell of the deceased

AN ETERNAL SECOND

There's a quiet comfort in an old home. A floor that creaks exactly where you expect feels like an old friend saying hello. The walls tell stories that span generations — as I run my fingers over the dents and scars, I think back to a clumsy stumble, a child's tantrum, a dropped vase, and try to remember a dozen more stories from long before I was born. Conversation and laughter emanate from the living room, breathing life and joy into an empty night. This house is safe, this house is comfortable, and that is making it *so* hard to leave. I know that, when I do, I will have to say a goodbye. It's a goodbye I've been avoiding for my whole time here, because I can't bring myself to make it official. Unfortunately, tomorrow, I'm flying home, and I can't leave these words unspoken. If I do, they will linger and fester, a poisoned thorn from a perfect rose. Ripping it out now is excruciating and it will leave a scar, but it's the only way that I can start to heal.

I take a deep breath, and try to capture one last second in this home. I take a mental picture: my aunt, so proud of the joke she'd just made, my mom and grandma lighting up the room with their smiles, wrinkles emphasizing their joy, the light giving them an angelic glow, the rest of my family nearly as bright and vivacious. I latch onto this moment of perfection, a comfort I will carry through a task that feels impossible. With great reluctance, I say "I'm going to see Grandpa Klaus now. I'll be back soon." Any response is lost to the deafening sound of my own heartbeat.

A blast of frigid air hits me as soon as I open the door. It's so tempting to return to the comfort of the house, but at the same time, as hard as this will be, I know it's what I need. As I take the first step out of the house, my mind starts to race. I imagine a hundred different ways to tell him how much he means to me. All of the words left unspoken, apologies I never had the courage to give, pieces of myself that I wanted to share but kept for fear of judgement or rejection. I try to find all the words, but there are simply too many, none of them right. I can't get this wrong. This goodbye simultaneously forces me to confront the tragedy of a loss, and threatens to sour something beautiful with an imperfect end.

I suddenly take in my surroundings, and realize that I'm almost there. I try the gate, and it is unlocked. I still have no idea what I'm going to say, but it can't be put off any longer. I close my eyes, take a deep breath, and realize I'm as ready as I can possibly be. I start to walk forwards, and stop when I see my destination.

A beautiful stone under an ancient tree that marks his final resting place.

Standing there, I realize the absurdity of all the worrying I've done in the past hour. I could plan a perfect goodbye and scream it into the void, but it won't be heard. I try to say "Hi Grandpa", but my words come out weak and strangled, barely formed before fading into the night.

I know that he isn't here, but at the very least, this spot gives me an anchor to my memories of him, it gives me a place I can call his. I'm by myself, but I don't feel alone. We sit in silence for a few minutes before I begin to speak.

A cascade comes, more than I had planned to say, more than I even knew existed. I apologize for not spending time with him, for all the most embarrassing and hurtful things I've ever done, I apologize for not saying "I love you" more often. I tell him all the things I wanted him to know about my life that I never got to share, my dream to be a professor and the role he played in it, my insecurities and fears, my friends and partners, my *life*.

I know that it's too late for him to hear it. I know that I will never speak with him again, and that there were too many words I left unspoken. But, as I think back to this, I think back to all the words I did say. The speech I gave at his 88th Birthday, where I told him two of our interactions that have shaped my life, his 90th birthday where I told him how much I loved and admired him, and a hundred nights spent in a beautiful backyard under the stars, hearing tales of a life longer and fuller than I can possibly understand.

As these memories come back, my words start to slow, and as my grandfather and I return to our silence, the cemetery reclaims the somber serenity it carried before my arrival. The tree, the flowers, even the grave itself are all calm and peaceful. This is a tranquil spot, one where he can get the undisturbed rest he deserves. Before I go, I interrupt the silence one last time

"Goodbye Grandpa. I love you, now and always."

When I go back to the house, I see a scene much like the one present when I left, but this time, I start to spot the cracks around the edges. A smile that ends just a little too soon, eyes that betray heartbreak, a heavy silence when the laughter ends. None of us are denying our grief; this joy is not a hedonistic distraction, but rather, a natural result of the acknowledgement of our shared loss.

I was terrified that, when I said goodbye, I would be heartbroken beyond what I could handle, that the reality of his death would hit me like a ton of bricks. Instead, finally sitting down and saying what was unspoken reminded me that it is *indescribably* lucky that we all got to have him in our lives. The pain of this loss is a price which we pay for the immense connection and love we shared, and it is a price we would all happily pay again.

How lucky we all are, to have something that makes saying goodbye so hard.

Golden

In loving memory of Klaus Foppa (August 1930-July 2021)

COOL DATE IDEAS WE THOUGHT OF IN AN HOUR

FOR USE AFTER OBTAINING ONE OR MORE SO'S USING PICKUP LINES FROM OUR ARTICLE IN mathNEWS 148.5

- ymh
- dc silent study in the same cubicle ☞ ☜
- couples that grind together stay together ☺
  - jazz room
- a little pricey but a really good experience if you're into live jazz
- 133tc0d3 d4t3 (hard problems)
  - make it a competition to see who can code a solution the fastest! may result in break ups
- lazeez (at least 5 lines)
  - competition to see who can eat the most lines
- washroom date
  - typically after lazeez date
- grand river rocks—rock climbing
  - rock her world by going rock climbing ahaha :weary: :lipbite:
- the waterloo location only has bouldering but the kitchener location has both bouldering and rock climbing
- mock interviews
  - "show me how badly you want this job ;)"
- ION to uptown + ice cream/bingsu
  - the roof on the parking garage has a nice view of waterloo
- ML 351
  - typically after lazeez date
- hide and seek while video calling
  - geoguessr but irl
  - don't get ditched tho
- crock a doodle
  - a little pricey but a fun experience if you're into painting/pottery
- duo queue bot lane inting
  - rakan xayah for bonus points
  - ezreal taric for more bonus points
- princess cinemas
  - nice venue for a classic, more intimate movie experience
- escape rooms
  - test your collaboration skills
- cook for each other!
- food ideas:
  - from personal experience, the cheapest "meal" at uni plaza is a junior burger with no cheese or bacon (around 2.50) $
  - ken's sushi $$$
  - chung chun $
  - lanzhou noodles $$
  - roosters on thursdays because they have a 3.99 chicken sandwich deal (not sponsored) $
  - d-spot 😳 $$
  - all bbt (chatime $$, coco $$, nowtea $$, the alley $$, yifang $$$, sweet dreams $$, gongcha $$, machi machi $$$$$)
- explore uw tunnels and bridges
  - try to get from one building to another without going outside
- geese catching
  - masterdaters are not responsible for any emotional or physical damages
- place a coin on the ion tracks and let the ion flatten it
- cry together because you are in cs
- cry together because you are in waterloo

masterdaters

N THINGS I THANK MY BEST FRIEND FOR

- Teaching me how to debug code using print statements
- Doing silly things with me
- Making me feel less silly and awkward
- Kindly reminding me that it's unhealthy (borderline dangerous) to eat 5+ chocolate bars in a day
- Always listening to my nonsensical and trivial rambles
- Reminding me that I am not the only one who thinks "I am the most boring person in the world"
- Going on weekly grocery trips
- Going on walks
- Trying out new food/recipes together
- Snack/drink recommendations
- Telling me fun stories about tropical islands
- Pointing out interesting observations and remarks
- Goose-watching, vending-machine-monitoring, and mars-bar-counting
- Reminding me that imposter syndrome is real
- Patiently teaching me fun stuff from the CS courses I didn't take
- Patiently reminding me the theorems I forgot
- Reminding me to check documentation for CS 350 assignments (I would not have passed otherwise)
- Being a "proof by example" that good people exist in this world
- Many many more…

It is so lucky to be your friend. You are one of the most interesting and kind people I know.

Dedicated to my best friend FZ.

Happy (almost) Graduation! You made it!

evilevievil

VUCA GIRL

TO THE TUNE OF "MATERIAL GIRL" BY MADONNA

[Verse 1]
I'm volatile, beneath the smile
Pretending it's OK
For once in life I'd really like to have things go my way
Uncertainty surrounding me
How can I see the light, that's right
When there's no promise anything will turn out alright

[Chorus]
'Cause we are living in a VUCA world
And I am a VUCA girl
You know that we are living in a VUCA world
And I am a VUCA girl

[Verse 2]
All things complex, serve to perplex
How'm I to understand?
I'm barely holding on just to keep up with life's demands
This place unjust, ambiguous
And I really don't know, oh no
How to keep on persevering in this world of woe

[Chorus]
'Cause we are living in a VUCA world
And I am a VUCA girl
You know that we are living in a VUCA world
And I am a VUCA girl
Living in a VUCA world
And I am a VUCA girl
You know that we are living in a VUCA world
And I am a VUCA girl

[Verse 3]
Yet with vision, and understanding
We achieve clarity
Adaptable and agile—this is the way to lead

[Chorus]
'Cause everybody's living in a VUCA world
And I am a VUCA girl
You know that we are living in a VUCA world
And I am a VUCA girl
Living in a VUCA world
And I am a VUCA girl
You know that we are living in a VUCA world
And I am a VUCA girl

Finchey

UNLOCK THIS blackBOX WITH mathNEWS++!

ONLY ¥5 PER INTEL I9-13900K CLOCK CYCLE

CRAZY TITLE

This is the story of how I almost missed the last **mathNEWS** prod night of the term. You know how slow but steady wins the race? Well I made that mistake. Not that I procrastinated my departure to **mathNEWS** not at all. No, I ran. Big mistake. I didn't know about the risks. Went for a run with friends.

The four of us braved the blizzard and the effort, made it past a park, and then another park and that was all fine with me. But then at some point two of us ended up a bit ahead on the others and after that turn, they just disappeared. So we ran back and forth, desperatly trying to find these two poor souls we had lost on the adventure.

At this point, something really strange happened. My friend blew a whistle and his pet goose flew from the blazing sky overhead with the wings of a hundred electric cars. Together, they took flight and fulfilled the destiny of the one and only, true hero of the bridges. I've never been so entranced. It was something terrible. It was my first time trying root beer. I've never tried root beer before. We had taken a warm shower in the meantime. PS: If you are one of those "friends" who abandoned me on that run and lied to me, may the geese hate you. Outside of Germany, and maybe outside of Austria, everybody hates them. I didn't register for pizza. Fuck.

OnOfThoseCrazySportsEnthusiast

MATHSOC CARTOONS EPISODE 35 AND 36

MEAN VALUE THEOREM AND TAUTOLOGIES, CONTRADICTIONS AND CONTINGENT PROPOSITIONS!

Enjoy Episode 35 and 36 of the MathSoc Cartoons series: *MATH 137 — Mean Value Theorem* and *CS 245 — Tautologies, Contradictions and Contingent Propositions*!

Want to see the next comic BEFORE it's released and provide feedback to help us out? Sign up to be a reviewer at `https://bit.ly/cartoons-reviewer-join`!

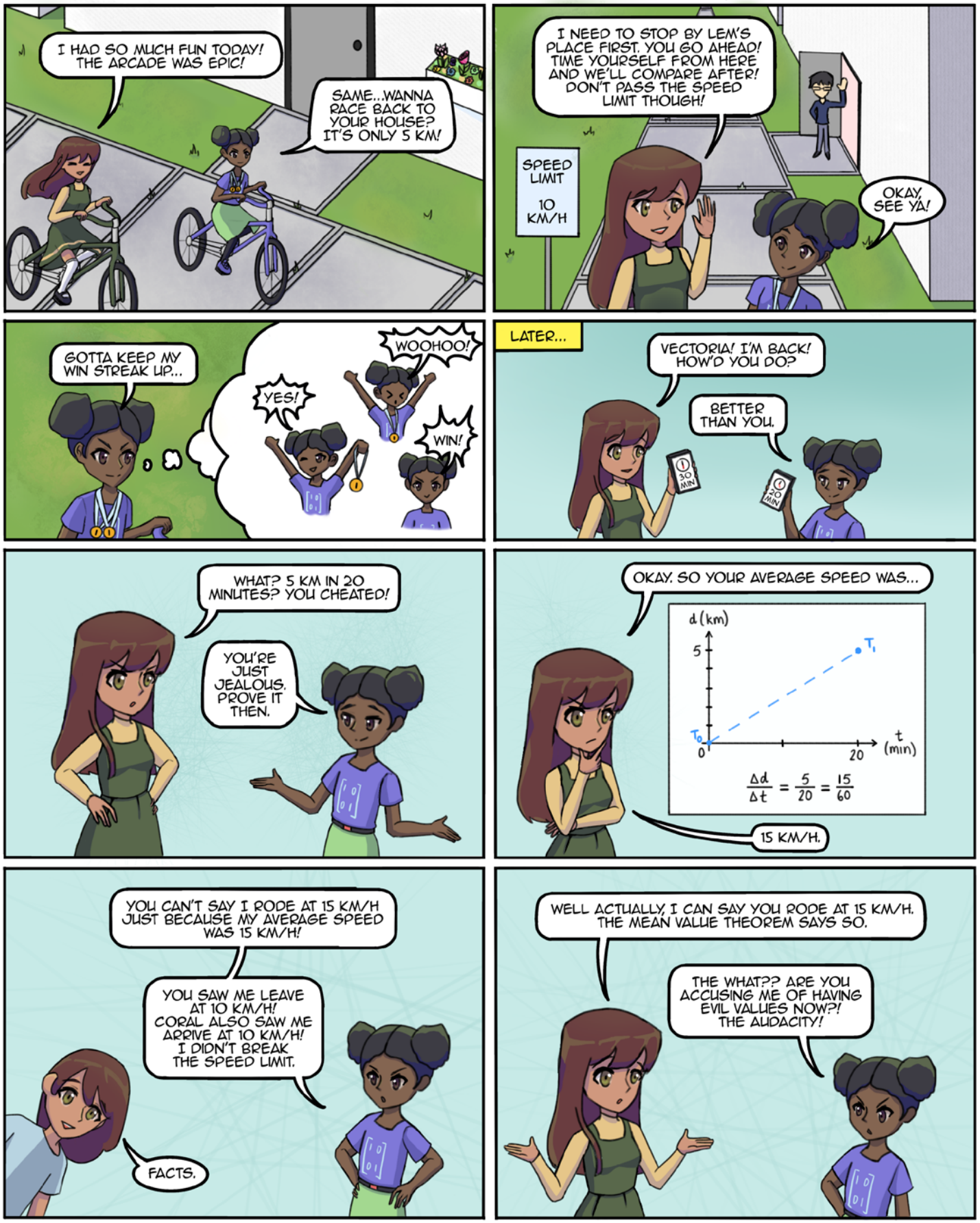
Additionally, applications to be a writer or artist for MathSoc Cartoons have been extended until April 6th! Check out @mathsoccartoons on Facebook and Instagram for more details and more comics!

As always, feedback, suggestions, and fan art can be left on the MathSoc Cartoons discord channel in the MathSoc server. We'll see you next term!

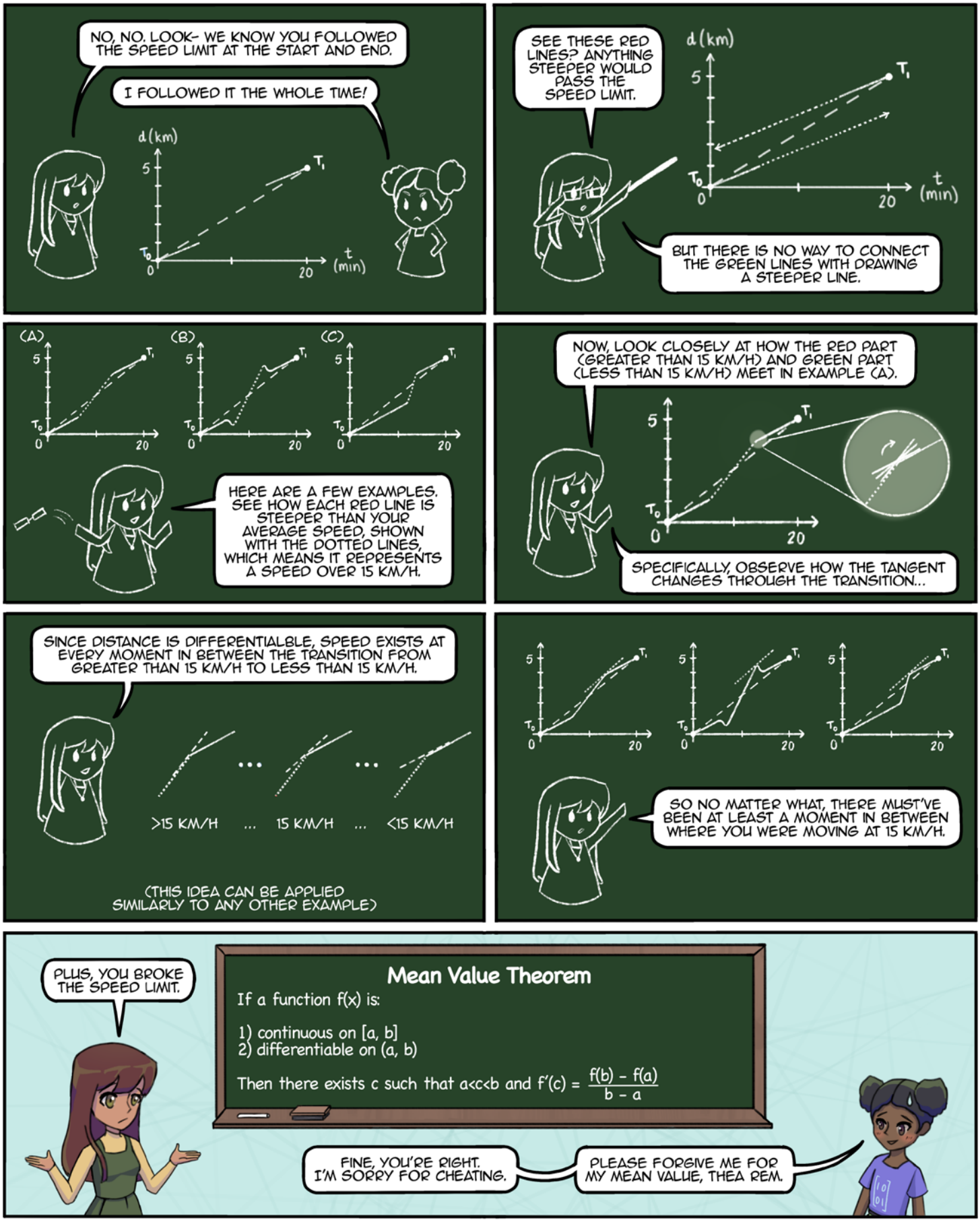
MathSoc Cartoons

MATH 137: MEAN VALUE THEOREM

STORY BY: CAITLIN KWAN, ART BY: WINNIE PHUNG



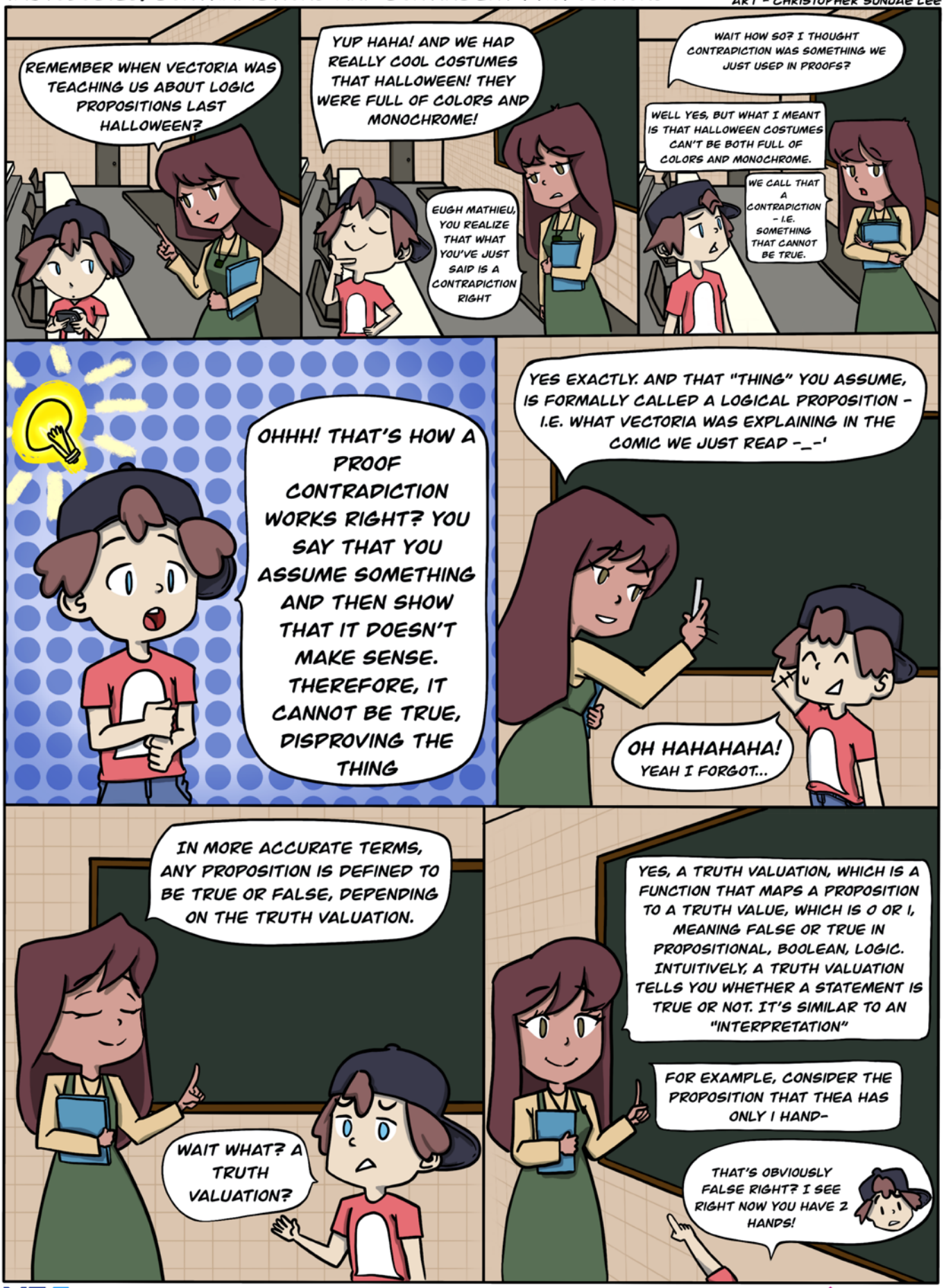
© 2022 Caitlin Kwan and Winnie Phung, all rights reserved. Published under licence by MathSoc. Do not reproduce.



© 2022 Caitlin Kwan and Winnie Phung, all rights reserved. Published under licence by MathSoc. Do not reproduce.

TAUTOLOGIES, CONTRADICTIONS AND CONTINGENT PROPOSITIONS

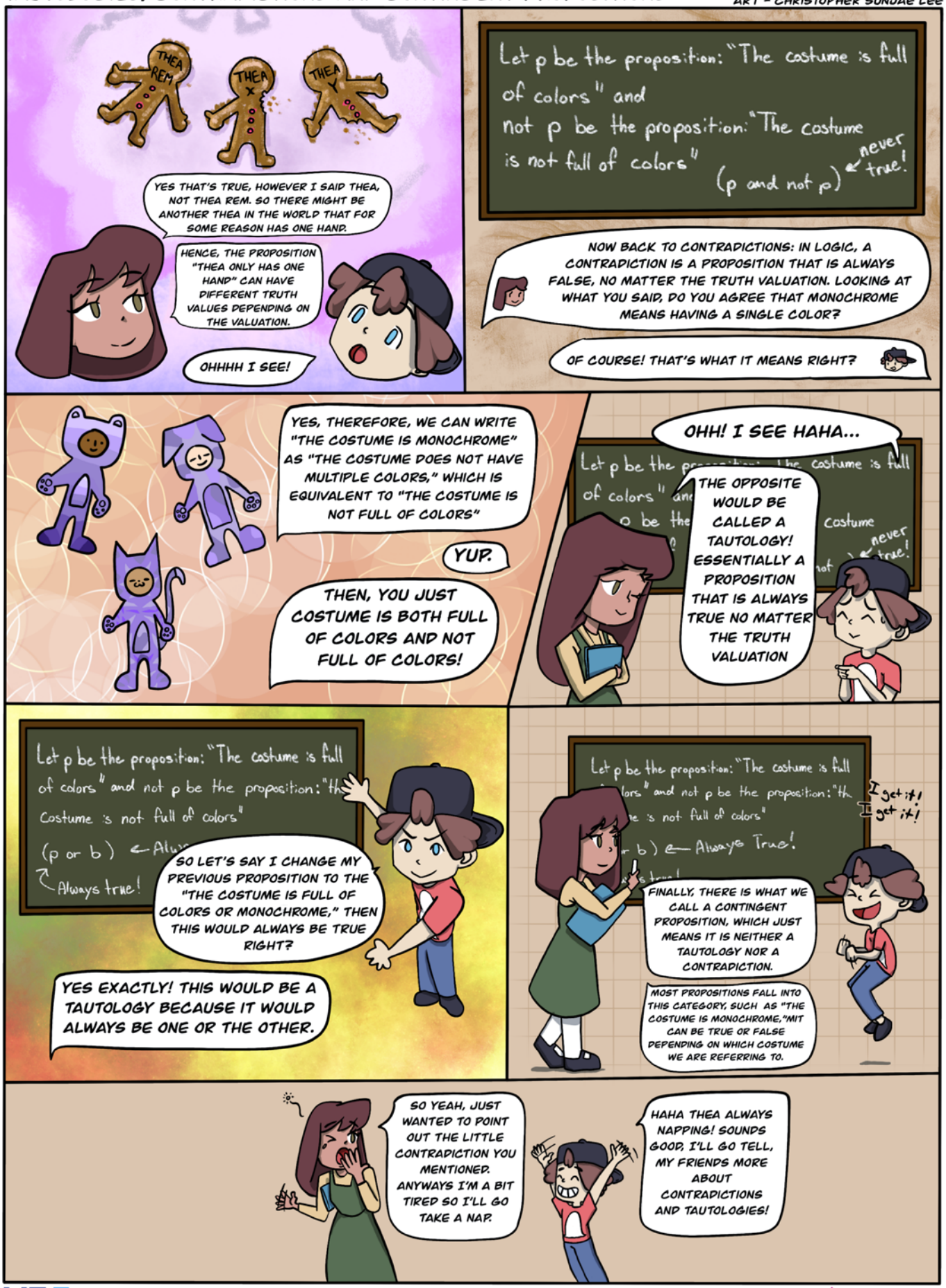
STORY - JIA FENG YU
ART - CHRISTOPHER SUNJAE LEE



© 2022 Jia Feng Yu and Christopher Sunjae Lee, all rights reserved. Published under licence by MathSoc. Do not reproduce.

TAUTOLOGIES, CONTRADICTIONS AND CONTINGENT PROPOSITIONS

STORY - JIA FENG YU
ART - CHRISTOPHER SUNJAE LEE



© 2022 Jia Feng Yu and Christopher Sunjae Lee, all rights reserved. Published under licence by MathSoc. Do not reproduce.

ALBUM REVIEW: KANYE WEST, 2016 TO PRESENT

ON WILLINGLY BEING GASLIT BY KANYE FUCKING WEST

It is the year 2020. Your name is Kanye Omari West, and you are a very active Twitter user. You look up from your phone and drag your finger down the large parchment scroll of problematic and concerning things to say next. The ink dried yesterday. However, you feel that today is a different kind of day. You freehand-type the following tweet.

> *YZY GAP BEGINS*
>
> *MOWALOLA JOINS AS YZY GAP DESIGN DIRECTOR*
>
> *KSG CARTOON MURAKAMI CUDI*
>
> *JESUS IS KING FILM ON APPLE JAMES TURRELL*
>
> *YZY SPLY DOC NICK KNIGHT*
>
> *FOAM RUNNER RELEASE MADE IN USA*
>
> *JESUS IS KING DR. DRE VERSION*
>
> *WASH US IN THE BLOOD VIDEO ARTHUR JAFA*
>
> *#WESTDAYEVER*

Without looking twice, you hit send. And you can't see it from the comfort of your mansion's basement, but all across America, the clouds open up and a bright light smiles upon the nation's most populous cities.

The subreddit r/WestSubEver is created.

Finally, there is a haven where Kanye West fans — the overwhelmingly white and suburban and male and teenaged pests that they are — can flock and honk words of praise for Kanye West (remember: that's you!). Regardless of their genders, they collectively identify as "WSE boys."

But notice that a crucial gene, thinner than a hair, has just been baked into the cake: delusion. Your fans treat your word as gospel. This isn't helped by the fact that you compulsively make promises that are clearly very unrealistic and unattainable. But the WSE boys hang on anyways. The sheer hope they feel with each release date announcement is only matched by the crushing despair they feel when the date passes with no album. You can hear the manic ruffling of feathers outside your walls, but this does not bother you. After all, you have made several of the greatest hip-hop records in recent memory. You put your phone down, pick up an Xbox controller, take a swig of root beer, and sink into your Louis Vuitton couch. What a morning.

❦

I've been struggling to write this review. Earlier today, I tried to discuss some ideas with a friend, but as words left my mouth, I would realize how insane they sounded. I felt inarticulate and irrational, honking and grunting my way through a skeleton of a review.

I felt like a WSE boy.

"I'm telling you, I can't put it into words, but there's just something magical about these Kanye records."

"How so?"

"Like…"

And I scrambled to pinpoint that surreal feeling I get from my favourite current-period Kanye albums. I brought up his use of hard cuts — attaching two audio clips side-by-side without a smooth transition or fade-in — and how these cuts allow him to destroy consistent senses of space. And if I had more time, I would have brought up the way his vocals are recorded and treated, and how they often contain audio editing scars that suggest that many freestyle tracks were spliced together, and how sometimes his words are so closely mic'd and so loudly mixed and so lacking in natural reverb that they feel as if I am saying them myself. And I would have said that these artefacts (alongside many, many more) suggest that Kanye makes all music in short, quick bursts. I would ascribe this decision to intention on his behalf, an intention to incorporate the raw and the imperfect — the most simple and brutal and intuitive musical decisions, the distillation of lifetimes of thought into seconds of action — into his perfect conception of hip-hop.

"I don't know anything about his recording process," said my friend, "but without this context, wouldn't it also be reasonable to assume that Kanye is incentivized to churn out the most amount of music in the least amount of time, all while managing his mental health problems, and that this may force him into these brash decisions?"

I couldn't argue with that.

"But also, his album covers are just…"

And I scrambled to pull up the cover for *The Life of Pablo*: a plain orange square with bold, boxy text and two crudely-pasted rectangular photos, one of a model and one of a church setting. It very well could have been made in MS Paint. And I try to explain that I've seen a lot of fan art online for this album, and that they have common features: grainy filters, epic photographs, Picasso paintings, paper textures. They are all pleasing to the eye, but none of these pieces of fan art are as bold or as attention-grabbing or as original as the actual cover for *The Life of Pablo*, which is conventionally very ugly. Nor could fans have predicted the album cover for *Donda:* a plain black square. And I try to explain that because Kanye imposes these strange covers onto us by declaring them as part of the final product, we are forced to live with them, which is how they become the original copies that all fan art merely reflects.

"Do you think that this was how Kanye intended for people to react to his covers?"

I don't know.

But I also don't think that it matters. All great artists die, and there will eventually be a day when Kanye will no longer be with us. In his absence, we will be left with his studio albums, these finished pieces of work that are no longer subject to spontaneous updates. They will be solid objects, and the listeners of the future will reckon with them, just as an archaeologist reckons with cave paintings, just as a child reckons with the leaves cemented in a sidewalk. Kanye will speak to us not through Twitter, but through his art. And I think that in this situation, his intention — what he did or didn't mean to do — matters so much less than the fact that he actually did it and that it makes some of us feel something.

That's what I find so amazing about Kanye. Through listening to his music, I become aware that I, as the listener, am just as central to the music listening process as he is, as the artist. I think that his music is energetic and inspiring and beautiful and abrasive and triumphant and heart-breaking and full of strife and perseverance and egotism and sadness and everything in between and not in between. And as someone that creates music, I see him as an art hero: someone that has made enough of a mess, that has broken enough of the rules to set me free. And most of what I've described above are projections of my thoughts onto his music, but that is exactly the point. That projection — a delusion — is an necessary part of engaging of art. To engage with art is to be a WSE boy: maybe not on the verge of mental collapse after each release date announcement, but definitely in touch with the thing inside you that can stare at an electrical outlet and make out a human face. It's about the benefit of the doubt: the kind I give Kanye is the kind that all art deserves.

❦

Another concern I had before writing this article is about the review format itself. When I write reviews, I purposely try to avoid saying things like this:

"*The Life of Pablo* is my favourite album by Kanye West. I like the way the synths tickle my ears, and I love the way soul samples are incorporated."

This is because when I place myself in the reader's shoes, I frankly kind of don't give a shit about what some stranger thinks about some album I'm probably not going to listen to. Plus, if I've written 1400 words about an album, you can probably assume that the album is somehow meaningful to me. So I've been trying to sidestep this issue by weaving stories around the music I review. I got this idea from John Green's *The Anthropocene Reviewed*, in which he turns reviews about human experiences like soccer games and disease into larger meditations on human suffering and compassion and progress. The goal is to focus not on the thing, but the stories around the thing. The thing is meaningless if not for the stories we tell about it.

It is 2:55 AM and I am getting tired. This seemed like the right review to say this in.

You don't need to listen to any of these albums. Kanye doesn't need to be special to you as well. I'm just glad I get to share my stories with you.

Thank you, and see you in the next volume!

χ

FUN HARMLESS MISCHIEF FOR APRIL FOOLS DAY

I love April Fool's day as I love mischief. Not everyone shares my love because some people's ideas of 'prank' are just bad. If the recipient isn't laughing it is not a good joke. I do think there are some very good pranks that just add more whimsy to people's lives. Here are some my favourite ideas:

- **Make a big announcement on social media that is actually true.** A lot of people are going to be rightfully suspicious about anything announced on April 1, so why not turn the tables and confuse everyone by announcing something true.
- **Put a hug me sign on your back.** This one works way better when there isn't a pandemic. Everyone who ends up hugging thinks they are the ones pranking you but — plot twist — this was your plan all along. It's a great way to get hugs when you are stressed. If you want to prank someone else, well you could also put a sign on their back. A particular **mathNEWS** editor still hasn't forgiven me for that one.
- **Randomly leave chocolate bars in people's mailboxes**. You will live rent free in people's heads as they will forever wonder about the mystery but won't complain about the increase in chocolate in their lives.

I hope you will use this day to add fun to people's lives.

Beyond Meta

THIS 4B'S UNIVERSITY BUCKET LIST – A HAIKU

gotta do it all
to avoid a null pointer
in my memory

mouse

HAZED 1

NO MATTER WHAT IT SOUNDS LIKE TO YOU, THIS IS FIRST AND FOREMOST A LOVE STORY

It was either raining, or,
had rained recently

the pavement shimmered
underneath the streetlights
glowing yellow and white

It was mid-way through
the fall term of 2019

We walked past the Burger King and the Onezo
(which would later become Machi Machi)

coming home from a **mathNEWS** prod night,
so it must have been around 10 p.m,
I turned to girafarig and said,

"Wasn't that hazing?
Hey, we got hazed, didn't we?"

"Uh-huh. We did."

"Did we really?
We totally got hazed!"

my voice echoed down University Avenue
through the quiet night.

(The safety training module for Scouts Canada, which I'd completed just a few months earlier, had really made it seem like hazing would be easy to spot.

It wasn't.)

Even as I shouted and squealed, *"We got hazed!*
That was hazing!",
as if it were some sort of major life experience I'd been waiting for all along,
I wasn't really sure myself.

Was that actually hazing?
Or
were we just thin-skinned
pissbabies?

❦

Essentially, Mao is,
or was,
a **mathNEWS** tradition.

It is (or was) often played
at **mathNEWS** prod nights.

Often described as a game with no rules.
Or, a game that you have to play first,
no explanation given.

"Have you played Mao before?
You have to try Mao.
Every new **mathNEWS** *writer*
has to try Mao. Go over there,
they're starting up a game right now."

That's how they get you to play Mao.
No explanation. No warning.

If you aren't familiar, Mao is a game where you get
yelled at,
laughed at,
and belittled. It is not a fun game,
and it is not a game that is designed to be fun.

I sat out, but girafarig joined. In total,
there were three new players, and two experienced players.
One was the DM.
The one who does most of the yelling.

❦

Risk. Before that I had been talking with a friend
about risk. Was I a risk-averse kinda person?
Did I take enough risks?

Was I ever gonna

(Coming to UW was in itself a risk. So was going to **mathNEWS**, to my first prod night back in 1B. My friend said
she took a lot of risks because she often went on Tinder.)

make it,
if I didn't take enough risks?

I knew I didn't take a risk that day. I picked
the safe option, didn't play Mao,
made it out alive.

❦

Mao is a card game of the shedding variety: the goal is to
"shed" all of the cards in your hand, and if you get something
anything
wrong,
the DM belittles you and gives you an extra card.

That's it. That's the whole game.

I will say the air was thick.
Though thick with what, I can't really say.
Apprehension? Unease? That sinking feeling
that comes with realizing that you hate
the people you've just met and now have to be
friendly to?

But faster than I had been anticipating,
the whole thing was already coming to its head.
girafarig put down his cards and said, *"Sorry,*
this isn't my thing."

"Yeah. No, yeah, no worries.
No, it's okay, I get it,
I get how it works,
it's just not something
I want to play."

I didn't know he had it in him,
to resign so coolly. I'd kind of half-almost-
expected more of a scene.

...And there was a scene
though girafarig had left by then

(And then, I'll never know what compelled this,
but a first-year who had been spectating with me
took girafarig's place instead.)

❦

In Sylvia Plath's time,
men were arrows and women were places
the arrow shoots off from. I
thought I got it, Plath,

I thought I got what it meant to be an arrow.
And I remember when I was the force
that propelled the two of us
to new heights.
(To UW, even.)
I took risks. I was spectacular
the world used to revolve around me.

But now girafarig's face scintillated
under the moonlight
as we held hands and braced
and wondered if we'd ever go to prod night again
and I was the place the arrow shoots off from.

I knew that I'd

never make it

but girafarig will

and just like that cursed game
of Mao, I would be there
to spectate

it

cy

ON ENDINGS

Imagine you knew this was the last **mathNEWS** prod night of your life. What would you do? Would you leave that essay you should really get started on for tomorrow? Would you rush to get to STC 0010 so fast you ask a custodian for directions because you can't afford the time to look for a map? Would you write an article that truly meant something, and ignore that pit in your stomach that something *just didn't feel perfect?* I think there's a person missing.

❦

We're not bruised, they're just party tattoos. And that colourful mess is just colourful regret.

❦

But I won't ever regret this choice that brought me here to the last **mathNEWS** prod night. I want to say goodbye. No, that's not what I mean. That's not what I mean at all. What I mean is, I want this to be perfect. I'm not perfect. *We're* not perfect. But if a goodbye is what it takes for a late night snowy trip to Farah's to get mushy tomatoes, where the plastic light blinks off the snowflakes on your silly red hat you pulled too far down over your eyebrows, to be memorable, isn't it worth it?

Why is it bad that I don't want to argue and fight and things to end in flames? Maybe I don't want to scar your memory with seething hatred. Maybe I wanted to do you the honour of being perfect, for a moment. Maybe I want to feel perfect, at this last **mathNEWS** prod night. And isn't that okay? Can't I let myself?

Do you hate endings too? Movie endings, book endings, school endings — my soul cries a little every time. I think no one really knows what to do with endings. She's scared of endings, she dreads them. He ignores endings, he jumps to flirt with anything that moves. They avoid endings, they hold on so tight they choke themselves both.

❦

But it's not the end, I promise! Time will move, it will move, it will *move* dammit. There's something behind that wall, after the last **mathNEWS** prod night. It's not a black hole, it's not four-eight-twenty months squished into a minute squished into a second squished into a moment until I can see you again. There's something behind that wall, and that's the rest of my life. Don't be scared.

❦

I'd like to see my mom now.

Absolutely smitten

-[----->+<]>+++.

It's the end, so let's begin. We'll read in some code, then write some code to evaluate that code. That's the plan, anyways. Though that's always been the plan, hasn't it?

❦

I know when I began, sometime over the winter break. But I'm not sure why I began. I think I had stumbled across someone raytracing a sphere in Desmos. Not sure where from. It was inspiring, in some strange way. I was probably procrastinating. So I began. Most probably procrastinating.

❦

A Brainfuck program is one of `+,-.<>[]`, or ASCII 43, 44, 45, 46, 60, 62, 91, 93 respectively. In addition, we will make use of `*`, or ASCII 42. Our first goal will be to read in a sequence of characters that make up a Brainfuck program, terminated by a `*`. For example, we want `+]*[` to look like `1 49 0` in memory: every character is offset by 42, and nothing after `*` is read in.

Brainfuck tip number one: always leave space. We start with `>>>>>`, for five empty cells. Will we need all five? Probably not. But I'd rather have indefinite waste than definite regret. Now, we'll read in a character, open a loop, and subtract 42: `,[----------------------------------------`. We'll give the cell to the right a value of 1: `>+<`. We call this our else flag.

❦

I've had this planned since. Start off with Desmos. Then LaTeX, code I had already written months before. Three slots to be filled in between. Then end off with Brainfuck. That was always the plan.

❦

Consider a scenario where we're reading in `-+`. Our memory then currently looks like `0 0 0 0 0 |2 1`, where `|` indicates the location of the pointer. Now, as the character we just read in minus 42 is nonzero, we're going to read in another character and shift right: `[>,>]>`. Our memory is now `0 0 0 0 0 2 43 0 |0`. Brainfuck tip number two: loops act as if statements if you end on a 0 cell. Notice how we've overridden our else flag. For now, we perform a mystery piece of code: `[→]`. As the current cell is 0, the loop is skipped. Nothing changes.

Suppose now that we were instead reading in `*+`. After subtracting 42, our memory is `0 0 0 0 0 |0 1`. But now, `[>,>]>`, the code to read in another character, doesn't trigger. Only the final right shift occurs, leaving us with `0 0 0 0 0 0 |1`. Right on the else flag. Time for the mystery code: `[→]`. It will wipe our else flag and shift one right, leaving us with `0 0 0 0 0 0 0 |0`.

But notice how in both scenarios, our current position in memory is two cells right of the previous character read in. So, we shift two back and terminate the loop: `<<]`. If that previous character isn't 0 (that is, isn't `*`), then we go back to subtracting 42 at the start. Else, we're done, and no more characters will be processed. Brainfuck tip number three: have loop invariants. Whether or not we enter a loop, we should expect the pointer to be in the same position afterwards, relative to the data we care about.

❦

Plans go awry. `s///` was never supposed to happen. Coq didn't even come close to working. But here we are.

❦

Of the 877 bytes that make up this program, our discussions thus far have explained only 65, 42 of which were reserved for consecutive minus signs. As such, while I find the intricacies in the rest of the code beautiful — admittedly, with some degree of bias — I will be unable to explain the rest of the procedure at such a low level. (Please absolutely approach me in person for the full story.)

Brainfuck tip number four: plan out the memory layout well beforehand. Our finalized layout will look as follows: `0 0 0 0 0 (ins) ast 0 cins tmp1 tmp2 ina dir 0 0 tmp3 cmem tmp4 0 0 0 0 0 (mem)`, where parentheses indicate an array of memory cells. We've already covered the first bit: `(ins)` contains the characters we read in, offset by 42. `ast` is the `*` read in — it's a 0 cell, but a special one. The nameless 0s are arbitrary in number, existing only to be useful spacing while debugging. The `tmp` cells are for executing small intermediary computations as necessary.

❦

I'm still procrastinating, by the way. I wonder if Marmoset takes Brainfuck.

❦

Let's zoom in on `(ins)`: if that array currently contains the instructions `1 2 3 4`, how do we know which instruction we're currently on? It isn't enough to store a memory cell `p` holding the current instruction position as a number. For one, what do you do if you have more than 256 instructions? More importantly, how do you tell Brainfuck to "move left / right `p` cells"? We can get a little clever. Let's look not at `(ins)`, but at `0 (ins)`, the 0 coming from one of the five cells we left superstitiously at the very start. Then, we say the cell to the right of this 0 is our current instruction.

Getting to the current instruction is now easy: `[<]>` loops left until it hits the 0, then moves right. Moving from `0 |1 2 3 4` to `1 0 |2 3 4` is also easy: do you see why `[-<+>]>` does the trick?

The same solution doesn't work on the (mem) array. Using a 0 cell to mark the current position only worked because (ins) was guaranteed to be nonzero. Now, we have to get clever again. Given memory `2 1 0 1 2`, we'll store it instead as `1 2 0 1 0 0 0 1 0 2`. Even positions contain the data. The current memory cell is the one to the right of the last 1 in the odd positions, or equivalently to the left of the first 0 in the odd positions. `[>>]<` gets us to where we want to go, and shifting left and right boils down to toggling odd position bits.

❦

The language itself is inspiring, in that same strange way. Purposely purposeless. Yet someone wrote the first "Hello, World!". Someone proved its computational completeness. God, someone made a 256-core computer for it. What is it like to be someone?

❦

Now, with the ability to get the current instruction and memory, we can copy them to the `cins` and `cmem` positions. This makes them easier to work with, as we now know precisely their relative distances from all of the machinery in between. Brainfuck tip number five: copying is a destructive operation. For example, if we want to go from `|3 0` to `|0 3`, `[→+<]` will do. But to go from `|3 0` to `|3 3` is more involved. We will have to double-copy with `[→+>+<<]` to turn `|3 0 0` into `|0 3 3`, then copy the second 3 back into its original position. This is especially tricky with the current instruction and memory, as we do not actually know their relative positions to where we want to copy them to. But it's doable.

With `cins` in place, we can handwave away the execution of `+-,.<>`. More or less, they consist of locating the current position in (mem), performing a basic operation, and then returning back to `cins`. Something interesting to consider: how does one implement something like that, a switch case statement in Brainfuck? And on top of that, keeping in mind tip three — no matter which case we enter, every path must merge into a single invariant position at the end. Especially considering there's an else flag somewhere in there to make it work at all.

Finally, we have to deal with the `[]` loops. The idea is that we maintain an inactive counter `ina` and a direction toggle `dir`. If we encounter a loop `[` and the current memory cell is not 0, then we execute instructions in the loop as usual. Otherwise, we increment `ina`. While `ina` is not 0, no instructions are processed. `[` continue to increment `ina`, while `]` decrements. Once `ina` hits 0, we will have skipped the loop. On the other hand, if we end a loop while `ina` is 0, and the current memory cell is not 0, then we must go back to the start of the loop. This is indicated by toggling `dir` and decreasing `ina` (i.e. setting it to 255). Later bits of code will read `dir` as an else flag to determine the direction in which to move the instruction pointer. Of course, among the other complications that arise, processing an else flag is, as we've seen before, destructive, and so even then there's work to be done. But you'll have to forgive me if we handwave.

So we'll take in as input a Brainfuck program. Then, an `*`, and then the input to the inputted Brainfuck program. And we are done.

```
>>>>>,[----------------------------------------
>+<[>,>]>[→]<<]>>>>>>>>>>>>>>>>>+<<<<<<<<<<<<<<<<<<[<]>[[-
<+>]<[→»[>]>>+>+<<<<[<]<]>>[>]>>>[-<<<<[-
<]<+>>[>]>>>]<<<<[<]<[→+<]>[>]>>>>>>>>>>[-
]>>>>>>>[>>]<[→<[<<]<<<<+<+>>>>>>>[>>]<]<[<<]<<<<[-
>>>>>>[>>]<+<[<<]<<<<]<<<<<<[[-<+>]<<+<----------
--------------------------------[--[>→<[-]<]>[→→«]]>[-
>+<]>>]<[→+<]+>[→«+>[-]>]<<[→»+<<]>[→»[-]<<]<+←[-
[-[-[--------------[--[------------------------------[-
-[>→<[-]]>[->>>>>>>[[-]<<<<[-]+←>>>>>]<<<<<<<]<]>[-
>>>>>>+>[[-]←→]<[-<<<<+>>>>]<<<<<<]<]>[-
>>>>>>>>>>>>>>[>>]+[<<]<<<<<<<<<<<]<]>[-»→
>>>>>>>>>>>>[>>]«→<<[<<]<<<<<<<<<<<]<]>[-
>>>>>>>>>>>>>>[>>]<.<[<<]<<<<<<<<<<<]<]>[-
>>>>>>>>>>>>>>[>>]←<[<<]<<<<<<<<<<<]<]>[-
>>>>>>>>>>>>>>[>>]<,<[<<]<<<<<<<<<<<]<]>[-
>>>>>>>>>>>>>>[>>]<+<[<<]<<<<<<<<<<<]>>>[→«+<<<<<[<]<[-
>+<]<[→+<]>[>]>>>>>>]<<[→»+<<]<<<<<[<]>[-<+>]>]
```

❦

Maybe it's pride, but I don't truly think I'm procrastinating. Being purposely purposeless certainly doesn't feel effortless, let alone meaningless. Maybe it's immature. Maybe it keeps me sane. I don't know. But in the end,

❦

Theorem 6.1. Brainfuck is a legitimate programming language.

hu

Footnotes:

1. The trick for marking position in (ins) was taken from Daniel B. Cristofani's implementation, which also holds the record for shortest Brainfuck interpreter in Brainfuck, at just 423 bytes.
2. The loop implementation with `ina` and `dir` was taken from DPAmar's implementation on CodinGame.
3. copy.sh's online Brainfuck editor is a godsend. They provide among many things a special `#` character that forces a core dump, an invaluable tool during the inevitable debugging.
4. *Corollary 6.1.1. Brainfuck in Brainfuck is a legitimate programming language.* Indeed, our program can evaluate itself evaluating a "Hello, World!" program in just 36 milliseconds over 56 minutes.

blackBOXES, BUT IN 3-D.

NOW AVAILABLE THROUGH mathNEWS++.

ONLY 31.415 $GME SHARES PER SIDEREAL DAY!

GOOGLE-TRANSLATING RANDOM PASSAGES OF TEXT INTO OBLIVION

COURSE CALENDAR DESCRIPTION OF MATH 137

Mathematics 137 Laboratory, LEC, TST, TUT 0.50

Father ID: 006880

Statistics 1 Statistics and symbols

Costs and inequalities. Commands and restrictions. Introduce the permanent limit of the lizard. Define and map the most powerful solutions. The same is true of line and Newtonian works. There is a negative price tag and limit. The model uses Taylor polynomials and Taylor's notation uses the Big-O number. Report problems with your computer. [names: F, W, S; lines: V, V, V].

Status: 4U system and vector.

Antireq: Matthew 116, 117, 127, 147

Is available online.

CANADA'S NATIONAL ANTHEM

Oh, Canada!
Our home and home!
Patriotism is an order for each of us.

With a bright heart we see heights.
The answer is really strong and free.

From afar,
We are safe in Canada.

God has made our country prosperous and independent.
We are safe in Canada.

We are safe in Canada.

DESCRIPTION OF ION ON THE GRT WEBSITE

There are 14 main obstacles to ION.

Each train is about 32 meters (105 ft) long. Each rear has 60 seats, more than 4 seats and more than 140 comfortable seats.

Upload the number and the train station to every train station.

Ion servers are 19. All Sites:

- Read EasyGO platform cards for unlimited payments.
- Payment machine for quick and easy ticket purchase.
- Each area has its own anchor wall.
- Driver's seat.
- Mike's message display provides new information.
- Covers and boxes for extra protection from the sun, wind, rain and snow.
- Surveillance cameras and emergency calls for personal safety.
- The yellow platform warns passengers not to board the train. The black dock door indicates the location of the train gate for the safe entry and exit of passengers.
- The overhead line provides power for ION transport.

WHAT THE PROCTOR SAYS DURING PAC EXAMS

Good morning,

Try to open your ears and head. During the last attempt, just drink or drink a clean bottle. If there is a mark on your water bottle, remove it immediately. If you want to take a short break during the trial, raise your hand and ask the lawyer to take you with you. Food and alcohol should be kept under the table.

Remove all tools and place in a basket under the table. You may not use a cell phone or other electronic device, including a smartwatch, during the test. Only pens, pencils, erasers and special items are on your desk. Everything else should be on the table. Comes with a pen and bag. If you need to provide personal information during the interrogation, raise your hand and get permission from the prosecutor. The bag should be tied and placed under the table so that it is not locked. Take the hat and place it under the table. If you have a hood, you cannot wear it during the test. If you have any questions, raise your hand and let the sympathetic nurse help you.

Put the Wat card on the board and fill out the candidate card. If you do not have a candidate form on your desktop, you can send the application form online. Students are not allowed to leave the bathroom for more than an hour without rest. There is a work time at each entrance in front of the gym. The trial lasted for two and a half hours. If you leave the gym while others are writing, go slowly to the left or right door so as not to disturb anyone.

Success on offer. You can get started.

WIKIPEDIA PAGE FOR UNIVERSITY OF WATERLOO

The University of Waterloo (also known as Waterloo, UW or UW Water Law) is a public research university located in Waterloo, Ontario, Canada. The large park is located in Waterloo, a 404-acre (998-acre) waterfront, and in Waterloo Park. The university maintains three members and four faculty. The university offers courses offered by law schools in six schools and 13 colleges. Waterloo has established the world's largest corporate education program, with more

than 20,000 students enrolled in the university's corporate education program. The University of Waterloo is a member of U15, one of the top schools in Canada.

chi-squared distribution

N TRUTHS AND M LIES - ABOUT-TO-GRADUATE REFLECTIONS EDITION

<heresy warning>

- I bought only one Waterloo shirt, and it's 'Faculty of Health' teal
- I actually read the textbook for CLAS 104
- I never got an email about booking grad photos
- I stole pink pens from the math open house for my mom
- I have an owl in a graduation gown to stave off senioritis
- I signed out a book from our library for the first time this week
- I procrastinated on my minor's 'statement of interest' for 3 years but got approved 15 minutes after submitting it
- I ditched salsa class to write this
- I am at in-person **mathNEWS** production night for the first time!

mouse

A PREFACE TO GAMER GAMES

Hmm... I need to start writing ahead of time, instead of relying on the narrow (and sleepy) window of time between prod night and noon the next day, the article submission deadline. The article I've been working on, gamer games, takes a look at the games I've played throughout my life. Part of why I decided on the topic is to examine how much I underestimate the importance of games in my life. Turns out the answer is something like "a lot", since I have way more to write than I realized. I guess gamer games will have to wait for volume 149.

Anyway, if you're feeling pretty gamer yourself, consider buying a subscription to **mathNEWS**++ at mathnews.uwaterloo.ca—it's for the most gamer of readers and only for the most gamer of readers.

cutlet

ON WORK TERM REPORTS

Work Term Reports. They exist, we write them, they mark them. It's a time-honoured tradition of co-op students. I was working on my Work Term Report for this term and I really got to thinking about the concept of a Work Term Report.

It's a very one-size-fits-all requirement. To ask every student to write a well-researched report thousands of words long about their co-op experience just doesn't make sense for a variety of jobs. Honestly, I feel kind of bad for a lot of the Work Term Report markers, especially the ones for Math, because so many of the jobs are similar in all but the details that can't really be shared without breaking confidentiality. It must lead to so many similar reports that all need to be written, read, and graded.

And for what?

There is very little that I've learned from a Work Term Report that I haven't learned from my job. One could argue that it teaches report-writing skills, but those aren't required in all that many positions, especially within Math. Besides, we already have to take Communications courses. One would think that those would cover this requirement.

Of course, there is another argument in favour of Work Term Reports, that they ensure that we've learned something over our co-op terms. This one is probably the best argument I've heard for the reports, and there's some merit to it; to make sure that students don't just sit at a desk all day while employers give them a rubber stamp because they don't want their interns to badmouth them to other students.

The only issue with this is that I could have written most of my Work Term Reports without actually having started my work term, with only the knowledge of what my position would be. Seriously, there is very little that I have actually picked up at my work term that I have then applied into a Work Term Report outside of what my research for the report already told me.

I'm not sure whether this is just me, but the fact that I'm able to pass my reports at all despite working in so little from my term kind of means that the argument revolving around your learning on your Work Term Report doesn't really hold out.

Basically, I don't see a good argument for Work Term Reports. I'm glad they're supposedly being phased out soon, but for this term at least, I will continue plodding along on my Work Term Report, writing thousands of words, until it is, at last, complete.

Predap

AFFIRMATIONS FOR THE MODERN WATERLUVIAN

You need to believe in a framework upon which you can make decisions/have fun/get mad at people/victimize yourself/claim virtue over others. Let manifestations provide a solid basis for your framework. Manifestations like these:

see @affirmationswaterloo on ig for more

cupcakeeater83

TOP 10 MODERN FAMILY QUOTES

Let's look at the top 4 *Modern Family* quotes:

> *What could be more natural than your mother's tongue in your ear?*
>
> GLORIA PRITCHETT

> *Sometimes, When You Read To Me, I Pretend To Fall Asleep So You'll Go.*
>
> LILY PRITCHETT

> *Here's the deal. Girls don't go for all that romantic stuff. They go for power and success, and since you don't have either one of those things… you're gonna be the funny guy.*
>
> JAY PRITCHETT

> *I had to actually come out to my dad three times before he acknowledged it. I'm not sure if maybe he was hoping he heard it wrong, like I said 'Dad, I'm grey.'*
>
> MITCHELL PRITCHETT

> *Success is 1% inspiration, 98% perspiration, and 2% attention to detail.*
>
> PHIL DUNPHY

Sparkly Moisturizer

RE: STATS BAD XD

What came first, statistics or combinatorics? The answer is obviously combinatorics, because as the age old adage says:

"Stats is ass"

I think the Joker said that, or maybe it was Michael Jordan? Idk.

Imagine if the math department said you were required to take another stats class after STAT230 and STAT231. I would be what they call, angerie. Petition for MathSoc to cancel statistics especially after the whole proctoring fiasco.

Wink wonk

WHAT EVERYBODY GETS WRONG ABOUT FOOD!

There are some big misconceptions about food that everybody gets wrong.

Let's start with fruits: for instance, strawberries. Or, wait a second…is a strawberry actually a fruit? The answer is no. The small seeds on the skin of the strawberry are nuts. Therefore, technically strawberries belong to the category of nuts.

Speaking of nuts, let us talk about peanuts, which are not nuts at all. Isn't that nuts? They are actually legumes. Peanuts are edible seeds enclosed in pods, and therefore belong in the same family as beans, lentils, and peas.

After taking care of fruits and nuts, let us consider vegetables. Tomatoes are a lot of people's favourite veggies, but wait, they grow on a flowering plant and contain seeds and therefore are technically called fruits. It is the same for eggplants, which are technically berries.

What becomes clear here is that human like to classify things. However, not all classifications make sense, and objects can end up sorted in some weird classes.

Sluty Sailor over and out

ON ON

On on on on

BILINGUAL PROGRAMMING

Once I decided that I would try to write a bilingual program. That is, a program which compiles (if applicable) and runs in two different programming languages. Note that writing a C program which also works in C++ does not count. I decided that I would start with two languages I knew relatively well: C and Python. And in the grand tradition of programming, I decided to start with a hello world program. I wanted to create a file which, in both Python and C, would print the string "Hello World!". Now, Python and C work quite well together for one simple reason: C preprocessor directives and Python comments both start with #. With this in my mind, I was able to abuse the #define directive to come up with this program:

```
#include <stdio.h>
#define print int main(void){printf
#define pass ;printf("\n");return 0;}
print("Hello World!")
pass
```

You can try this out for yourself if you want. It works in both Python 3 and C 99 (and quite possibly other versions too, but I haven't bothered checking). The Python side of it is quite simple. The first three lines are ignored as comments. Then, the second last line does the printing and the last line is a keyword which does nothing. In C, the story is a bit more complicated. The first line is a simple #include directive to get access to the printf function. The second and third lines are where the interesting stuff happens. Basically, they transform the Python program into a fully functional, if terribly formatted, C program. This is why there is a pass on the last line: there has to be something for the preprocessor to transform into a valid ending to the C program.

After this victory, I went on to write some other Python/C programs. They were generally horrible looking and difficult to write. But then I wondered if I could do better. I wanted to develop a general method to write any Python/C program (or almost any). I eventually realized that any C program, minus the preprocessor directives, can be written on one line. This allowed me to do this: (the program contains some very long lines, so it may not be formatted properly here)

```
#include <stdio.h> // You can also include anything
else you need
#define program void some_func(void){printf("Another
function\n");} int main(void){printf("You can include
any C program on this line, even with multiple
functions.\n");some_func();return 0;}
#if 0
# The Python program goes in here
print("You can include any Python program here.")
program = ''  # so Python recognizes the last line
#endif
program
```

I think that this can be used to write any reasonable Python/C bilingual program (exceptions may include other weird types of programs, such as a quine). Having conquered Python/C, I decided to try Racket/C (I only know 4 programming languages, two of which are C and C++, so my choices are relatively limited). It took me a while, but I managed to create this hello world program:

```
; extern int printf (const char *__restrict __format,
 ... )
; int main(void) {
; printf("Hello, world!\n")
; return 0
;} /*
(define (print s) s)
(print "Hello, world!")
;; This is my way of pretending to print in Racket
;*/
```

The first line is there because Racket complains if you try to #include anything. Other than that, the C program comes first, but with a semicolon at the start of every line instead of the end so that Racket ignores it (yes, this is allowed: try compiling it and see). After that is the Racket program, but wrapped in a multi-line C comment so that C ignores it. I believe that this method is sufficient to create any reasonable Racket/C program.

The logical next step seemed to be to try Racket/Python. I tried to write a hello world program which both Racket and Python will accept. After nearly giving up, just a few minutes ago I came up with this:

```
'''()
(define (f s) s)
(define (print s) s)
;'''
(print(f"Hello World!"))
```

It works, somewhat. You see, Racket seems to complain whenever it sees a # character, so I tried to use a multi-line string in Python to make the Python interpreter ignore the Racket program. But Racket, as you may remember from CS 135, treats the ' character kind of like a function (I don't understand the exact details). So instead of printing "Hello World!" like I want it to do, it prints

```
(list 'quote (list 'quote empty))
"Hello World!"
```

This is not ideal, but at least it kind of works. However, I do not think I can generalize it to write other Racket/Python programs, or at least not easily. I would like to find a better solution, but I cannot currently think of anything.

If you like writing strange programs (as I'm sure some other writers here do), then you might enjoy bilingual programming. If you manage to get a better solution to Racket/Python, I would like to see it.

((λ(* /)(* * /))(λ(+ -)(if - -(+ + #t)))#f)

MINOR VICTORIES (UNLESS YOU'RE VINCENT)

On the heels of Vincent's bombastic work with defending the privacy of math students (and possibly some engineers) across campus through the new policy restricting use of invasive proctoring software, I'm inclined to also commemorate minor victories of the week, especially given the flurry of General Meetings that passed last weekend.

From positive changes in tuition rebates at Senate, to adoption of EDI stances in accessibility, representation, and decolonialisation, the work of students has helped create a more inviting environment for all. However, there is plenty of work left to be done. At the Engineering Joint Annual GM, which I had the opportunity to observe, many a policy was left on the table due to an early loss of quorum, and major and possibly horrendous changes are about to be put forward at WUSA's GM the evening I write this article; the toil is far from over. Take faith those of you who hauntologise the UW Bike Centre and its now loathsome corpse—the future brings opportunities to revive what makes campus life complete, even if the form seems yet so foreign.

While we celebrate the major achievements, let's also toast the smaller quality-of-life changes, such as the one that prevents your great nephew from getting screwed out of his tuition refund because of some legal obscura. We might never meet, but that's what being altruistic is all about.

sqrt(cause)

CANNIBALISM IN MODERN DAY ATLANTIS PART 2.

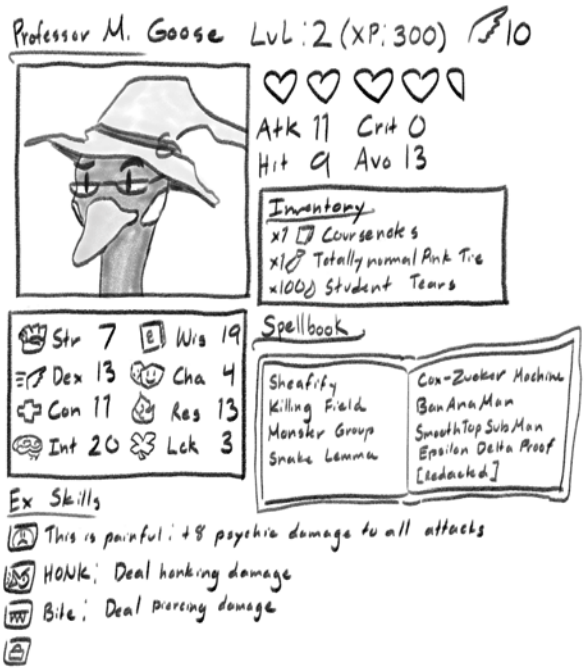
1535 left the populous destitute, as the carp required for fish fluid production suffered a bout of bacterial sponge syndrome. Bacterial sponge syndrome is, oddly, a fungal based disease that leeches the nutrition from fish fluid. First discovered in the blue-water carp in 1530, by 1535 it had spread to the deep-water carp and the sea carp, which in total comprised 75% of the fish fluid production supply. This famine led to a glut of salt-cured carp as millions of fish had to be slaughtered in the hopes of halting the spread of this insidious infection. Salt-cured carp would become a cultural staple for next 400 years.

1581 is defined by the emergence of socialism as a system of governance in New Atlantis. Coral Marx, regional head of the Sapien Liberation Front, won the general assembly vote with a 51% to 49% margin, narrowly beating out the Naga Politician's Guild. The NPG called for a recount of the votes, but since the vast majority of the voting body were weeks old shrimp larvae, they had died moments after the polls had closed, invalidating any attempt at a second count of the entirely verbal ballots. On the brink of losing their administrative control, the Chief Strategist for the NPG, Ser De Van Der Paulo proposed an unorthodox solution: split the city. If Atlantis were separated along party lines, there could be two different governments. This was quickly scrapped as this plan would take years to reconfigure the public utilities and shuttle the population based on an uncountable vote.

SecretSquirrel

THE ADVENTURES OF PROFESSOR M. GOOSE CHAPTER 5

Well, we're taking a break from the action, because there's nothing readers love more than a cliffhanger, right? Because clearly the story about Professor Goose, and for some reason multiple cat themed characters, needs more tension. So anyway, here's some insight into what our favourite professor can do:



Now it is currently 7 minutes from the already extended deadline so we're ending here. (Also I need to ~~lookup what half the words in the spellbook mean~~ do absolutely nothing because I always write about topics I am well-informed about, and also I am a real math student). There's nothing more here so

To be continued…

*Not a N*rd*

Hello fellow math god! Want to decide what happens to Professor M. Goose? Come to the next prod night or email your suggestions to professormgoose@gmail.com

GLORIOUS CANADIAN SOCCER

Canada has qualified for the 2022 FIFA World Cup!!! Celebrations all around for sure, but it is especially glorious if you have been following the team's growth and improvement in recent years.

The growth of Canadian soccer has been absolutely exceptional. Recent accomplishments include the men's team qualifying for the FIFA World Cup in a spectacular fashion for the first time since 1986, and the women's team winning the Olympic gold medal last year. In the organizational side, Canada Soccer's new professional soccer league called the Canadian Premier League is now on its 4th season, with teams good enough to even beat Canadian Major League Soccer teams occasionally. Canada will also host the 2026 FIFA World Cup alongside Mexico and the US.

Based on the stats, results and current standings of the team in World Cup qualifying, it appears it was all smooth sailing, winning matches with ease. However, it wasn't necessarily like that at the start of the qualification campaign.

❦

Canada's men's national team was never really that good. Canada wasn't a consistent feature in World Cup qualifying, often being knocked out in earlier rounds. In the Gold Cup, Canada usually would not make it far into the knockout round, or even make it at all.

In 2019, Canada beat the US 2-0 in the CONCACAF Nations League. It was the first time that Canada had won against the US in decades, making it a monumental accomplishment by the team, though some thought it was just a fluke. However, losing against the US in the return fixture meant that the Nations League finals were out of the picture for Canada.

❦

The next main competition after the Nations League was the start of the World Cup qualifiers, with a revised format. Had the original format stayed, Canada would not be qualifying automatically for the World Cup right now, but thankfully, those at CONCACAF came to their senses and drew up a more fair format.

CATCH MORE OF YOUR FAVOURITE blackBOXES
WITH A mathNEWS++ SUBSCRIPTION!

ONLY 20000000 ZIMBABWEAN DOLLARS (2022)
PER LEAP SECOND

In the first round, Canada was placed in Group B with four other teams, with only the top team proceeding to the next round after 4 matches. The first three matches, against Bermuda, the Cayman Islands, and Aruba were a breeze. They even set a new team record for biggest win of 11-0 against the Cayman Islands. That left Suriname for the final match of this round. I remember when the biggest thing to worry about was whether or not we could win against Suriname! Thankfully we won that match and proceeded to the second round.

The second round was simpler, with Canada needing to win on aggregate against Haiti, in two matches home and away. Again, people were more worried about this second round. Haiti brings back bad memories, especially of Canada losing against them in the semi-finals of the 2019 Gold Cup.

Canada ended up winning 4-0 on aggregate, helped by an amazing stroke of luck via a disastrous own goal by the Canadian-born Haitian goalkeeper one minute into the second half of the second game. Canada proceeds to the third, and last, round of World Cup qualifying.

❦

In between the second and third rounds of World Cup qualifying was the 2021 Gold Cup. Canada was expected to advance to the knockout rounds, and they did, placing second in the group just behind the US. In the 2019 Gold Cup, Canada was knocked out of the competition by losing to Haiti in a surprise defeat. Would Canada do better this time?

Thankfully, we didn't see a repeat performance, and Canada cruised into semi-finals against Mexico, arguably the number 1 team of CONCACAF. However, despite the monumental challenge facing the team, Canada performed phenomenally. Even with Mexico up a penalty goal at the end of the first half, Canada managed to tie the game 1-1 and kept it that way almost the way to the end. Canada sadly ended up conceding a goal 9 minutes into stoppage time, giving Mexico the win. It was a valiant effort against basically the strongest team of CONCACAF.

Who would have known then that this would be a glimpse into what more was to come?

❦

The third round began a month after the Gold Cup. It has 8 teams with the top 3 teams qualify automatically to the World Cup, while the 4th placed team advances to the inter-confederation play-offs. My hopes then were that we would at least place 4th for that play-off spot.

In the first series of matches in September 2021, Canada drew twice and won once. It didn't feel impressive, but the team was doing well enough to get at least 4th, especially with one of the tied games being against the US.

It didn't end there. In the next three matches in October, Canada again drew two games and won one. Similarly, one of the tied games was against the strong Mexico. It is considered extremely difficult to score against Mexico in their Estadio Azteca, and Canada managed to do it to end up with a tie game. Canada did not lose a match in Mexico!

The next two matches in November were crucial. Mexico was to play against the US, then against Canada. As these are the top 3 teams in the table, these two matches would help shape the path to the World Cup. Canada used its geography to its advantage here, by scheduling the matches in cold and snowy Edmonton.

The US won 2-0 against Mexico. For Mexico, that result made the next match even more important, but disaster waited: Canada won 2-1 against Mexico! This propelled Canada to the top of the table. Only in my wildest dreams would I have ever imagined such a thing, and yet it was in reality that Canada was first place!

Canada remained top of the table in the next series of matches, winning one against the US. This meant that Canada drew once and won once against both the US and Mexico, the two top teams of CONCACAF—I would not believe you if you told me this in September.

The last three matches were scheduled for March. All Canada needed was one win. Even with one draw, Canada would still be able to qualify, depending on other teams' results. Even losing all three left some scenarios where Canada would still be able to qualify. That is how dominating the performance of the team was.

Unfortunately, Canada suffered its first loss of qualification against Costa Rica 1-0. The next match was against Jamaica in Toronto, and by winning that match 4-0, Canada got their win, and qualified for the World Cup after 36 years. As soon as the match ended, it was like a weight was lifted off of my shoulders, and no doubt the shoulders of many others, as we all could relax again and start celebrating their accomplishment.

❦

Pardon the pun, but it really did feel like the goalposts kept moving more and more as Canada kept doing better and better. Let's recount:

– In the first round, it was whether or not we could win against Suriname to advance to the second round.

– In the second round, it was whether Canada could win against Haiti (who once shockingly won against Canada at the 2019 Gold Cup quarter-finals) to advance to the third round.

– At the start of the third round, all I could hope was that Canada would get fourth place and advance to the intercontinental play-offs.

– However, once Canada become first place in the table, the target was moved again to whether Canada would maintain their top three position and qualify automatically to the World Cup.

– Once it became clear that the chance of Canada qualifying automatically was around 99%, my main concern became on the seeding of the 8 groups at the World Cup.

– Now, after Canada secured qualification, I have one last qualification to concern with, which is whether Canada would qualify automatically as the top placed team out of the 8 teams in this round.

As of the time of writing, Canada has one last match away against Panama. Currently, Canada has 28 points, so even if Canada loses this last match, they will probably remain on top of the table. A disaster would need to occur for Canada to fall to second or third place.[1]

What's even crazier is that with the World Cup secured, there are even more targets to aim for. The last time Canada was at the World Cup in 1986, Canada lost all its matches and didn't score a single goal, thereby ranking 24th out of 24 teams. You can imagine what we hope to accomplish this time around, such as scoring at least 1 goal and not losing every single match. I am sure, as the matches start, that the objectives will keep moving up more and more.

❦

I wish I could include more about everything Canadian soccer went through, but I would probably end up writing an article the length of a small novel. I can't state enough how much I left out of this article. There is the coach John Herdman as a start. The wonders he has done for both the women's and men's national team can't be understated. There is also a lot to say about the individual players themselves. I could have also gone on tangents about other aspects of Canadian soccer, like the knock on effects from the Canadian Premier League for example. There will likely be another article about Canadian soccer where I inevitably go more into those topics, but for this article, I must cut off at some point.

Once World Cup qualifying ends, there will be some 2022-23 CONCACAF Nations League matches before the World Cup itself. I do hope that everyone does get a chance to watch the Canadian team at those matches before the big ones at the World Cup.

boldblazer

1. As a side note, Canada would have already been guaranteed first place in the table had they not lost against Costa Rica. While Costa Rica did win 1–0 against Canada, they did not win honourably, and I hope their conduct improves should they qualify for the World Cup by winning the inter-confederation play-offs. I do not wish any team at the World Cup to be on the receiving end of what Canada went through in Costa Rica.

A DEFENCE OF LIGHT MODE

I HAVE RISEN FROM DARKNESS AND SEEN THE LIGHT

Before I present my argument, I would like you to be aware about the Backfire Effect[1], which is a well-documented psychological effect about how humans are averse to entertaining well-reasoned arguments that are in opposition to their current worldview. I do this because *every single time* I try to defend light mode, the argument around me devolves to some variation of me being wrong or cringe or just dumb, without any supporting evidence to back that up.

Now! Onto the cringe.

INTRODUCTION

Over the past N years, there has been an astronomical rise in the use of "dark mode" in websites and other computer software. From what I can tell, the earliest origins of dark mode are from cathode-ray displays in oscilloscopes and early computers that would default to a dark black screen, and then turn a phosphor layer behind the screen bright wherever a electron gun fired electrons onto it. Since it was probably less electrically expensive to have a gun fire just enough bright electrons to display some text rather than fill the entire screen except for the text with white, computers stuck to a black background with green/white text on it, at least until graphical interfaces became the norm.

THE TERMINAL ON THE LEFT WAS PROBABLY LESS EXPENSIVE TO DRAW ON A CRT SCREEN. *Conversely, it was probably more ink to print on.*

That's probably why a lot of developer tools, like terminals and terminal-based text editors, use dark mode—it used to be simpler to do it that way. Of course, now that we all use LCD displays, this problem is mostly a non-issue. But, in 2018, French designer Sylvain Boyer showcased a phone UI with "dark mode", allegedly with the dual purpose of looking pretty and saving battery[2]. This quickly caught on with developers, who were already using this colour scheme and had started to champion it, and soon you had dark mode code editors, then websites, then web browsers, then operating systems…within a few short years, everything became dark mode.

I don't think dark mode is necessarily the best idea.

Now, the proponents of dark mode make two arguments: first, that it is easier on the battery, and second, that it is easier on the eyes. Let's dissect this.

EASIER ON THE BATTERY?

Now, it is true that if you have an OLED display, your device will use less energy when displaying pure black images rather than pure white—this is because an OLED display controls lighting at the per-pixel level, and if the pixel isn't supposed to be giving off any light, it will quite literally just turn off. Note that this only applies if you're outputting *pure* black—if it's even a dark grey, that pixel is on (albeit at a low brightness) and a lot of the benefit of dark mode on an OLED goes away. Most dark modes, you'll observe, are dark grey/dark blue rather than black, because black is *really* hard to make look good. Some try with a "dark" and a "black (OLED)" option; I think most fail.

And here's the other rub—it's very likely that your devices *aren't* OLED.

The vast majority of laptops are not OLED—they can't be because of the screen burn-in issues that OLED has. An OLED pixel that displays the same image for a long amount of time will eventually "burn" the image into itself, and permanently get stuck on either that colour or a fainter version of it. So if you always have the Start menu on the bottom of your screen, or the Apple logo on the top of your screen, that image will burn into your display. And whenever you open something full-screen, the ghost of the tilted window or the half-eaten apple will be there, tormenting you for all eternity[3]. Recent technological advancements have made OLED more viable on laptops, so some newer laptops come with it, but *not* the latest MacBooks, *not* most of the newer ThinkPads, *not* anything good that costs less than 1,500 USD[4]. So not anything the vast majority of us are using.

The vast majority of desktop monitors are not OLED—for the exact same reasons as above. There are even fewer exceptions for this—unless you specifically know that you bought an OLED monitor, I'm willing to bet you 1,000 fake internet points that your monitor is not OLED.

A lot of new phones these days actually *are* OLED, but not the vast majority of them by a long shot. IPS is still the most common technology for budget phones, and OLED has only become more popular in recent years. If you use something cheap or old (as you might, being a university student), you're probably not using OLED.

So if you're not using OLED, what are you using? Most likely an IPS, VA, or TN display. What all of these have in common is that there is a singular common backlight for all pixels that is turned on at the same time. Regardless of the colour you're displaying, the pixel is lit and emitting light. No power savings here[5].

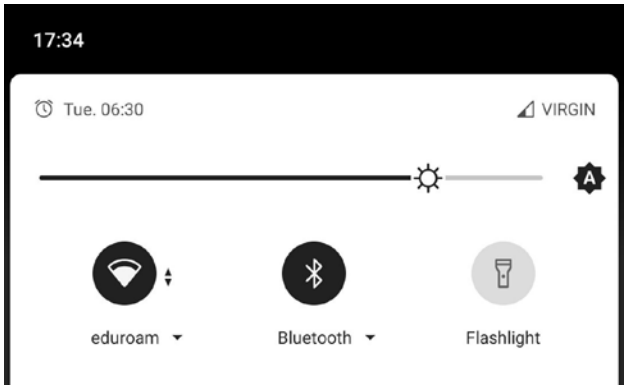
So for most of your devices, you're not getting any power savings. What about the other argument?

EASIER ON THE EYES?

Folks claim that it causes them less strain to read white text on black rather than black text on white, since the screen isn't blasting the light of a thousand suns at them. And if you're

one of those folks, I would like to introduce you to something revolutionary—something so simple, so brilliant, you'll be frankly surprised that no one's ever thought of this before.

Your screen has a brightness slider.



YES, YES, HAHA, MY PHONE PROVIDER IS "VIRGIN". *I made the obvious joke, now you have to interact with my argument.*

You can see that it's set pretty high right now—that's because I'm in a well-lit area where a brighter screen is appreciated. When I'm in a dimly-lit room, thanks to the virtues of automatic brightness, this fades to a very low setting (often even the minimum setting). If your auto setting is too high in dark rooms, you can move the slider and the "adaptive auto brightness" in most phones will recognize that you want it to bias a bit lower. Once you do this, eyestrain is eliminated even in the darkest of rooms!

Also—now your device is actually readable in bright light. Dark mode displays are *hilariously* unreadable in bright light, even at max brightness. Meanwhile, light mode manages just fine, at minimum brightness or a couple steps above it.

And you know the best part? On a non-OLED display, running your brightness lower means that the singular backlight we talked about runs at a lower brightness. This is by far the most power-hungry component of a non-OLED display, and you just significantly reduced the load that you put on it. That's right, light mode can in theory, and in certain circumstances, have better battery than dark! In my personal experience, the battery life on my phone (OLED screen) and my laptop (IPS screen) have both been unaffected since I switched from dark mode exclusively to light exclusively.

What else you got?

I JUST THINK IT LOOKS PRETTIER...

It's okay, we all have our own personal opinions. The important thing is to realize that dark mode is not *objectively* better—the preference is highly subjective, and just because someone else prefers light mode does not mean you get to shame them for having the correct opinion.

For a detailed analysis of why dark mode might not be the most aesthetically pleasing choice for all cases, including an in-depth analysis by famed UX designer Desmond "DesAiner" Ainer, check out the premium version of this issue, **mathNEWS**++ 148.6. Thank you once again to our premium subscribers, for these in-depth analyses of your incorrectness would not be possible without you!

CLOSING THOUGHTS

I find that a lot of the defensiveness for dark mode is driven by people who heard the battery life argument once, and then never stopped to reconsider their views. If you think that might be you, consider giving light mode a try for a couple weeks. Turn down your brightness, see if you can get used to it. If not, that's fine! At least you checked, and didn't eliminate a valid option without even considering it first. If you do come to love it, remember to shill it like I am doing right now. Remember, now that the Arch Linux users are becoming self-aware, we need *something* that we are in a tiny minority of and can be elitist about. Life is all about looking down on others, and I hope that by looking down on you today I have given you the tools to look down on someone else in the future.

[redacted]
(get **mathNEWS**++ *to find out this writer's real identity today!)*

1. https://theoatmeal.com/comics/believe
2. https://www.fastcompany.com/90160980/this-brilliant-smartphone-ui-saves-your-battery-and-looks-classy-af
3. Clearly, the solution is to just use Linux, since you'll never use the same DE long enough for UI elements to burn in.
4. I saw a couple listed for CAD 700–800 that had a Pentium N6000 CPU. That's a $200 laptop part. Sure, you could sacrifice "having a usable laptop" for "muh dark mode", but do you really want to?
5. Certain devices have "local dimming", where a part of the screen displaying pure-black can have its backlight turned off independent of other such parts. These are about as rare as OLED laptops, perhaps even rarer. The best counterexample I know of is the iPad Pro, I think.

SIMPS

You're in CS 146 right? Are you doing the simp thing?

That's what all the CS 146 people do! The simp thing!

str(n)cat

NO ONE LIKES BLURRED BUNNIES.

SUBSCRIBE TO mathNEWS++ FOR UNBLURRED BUNNIES.

ONLY £18 12S 09D PER LIGHT-SMOOTS

RE: OPINION: THE FEDBUS NEEDS TO GO

While the previous article mentioned important factors for getting around such as comfort and cost they miss a key aspect—and that is travel time. In the following chart it is clear that for all but two trips, York Mills and McCowan/ Scarborough Centre the Fedbus is over an hour faster. This additional time is especially stark for Aldershot GO and Richmond Hill where the Fedbus is twice as fast. Sure you're more comfortable on a GO bus but it takes forever whereas the Fedbus is direct with no stops.

There is also the matter of transfers to consider. At many transfer spots you're outside and in the cold most of the time from November–April. That is something that I, personally, would rather avoid.

I also must point out that the author mentions a bunch of physical perks of GO Transit stations, including parking, dedicated bus shelters etc, without noting that basically all of the Fedbus routes go to GO transit stations and thus are also able to take advantage of said infrastructure.

Furthermore the assumption was made that after getting to this transit spot the student would continue to use public transit to get to their end destination. If this assumption is true, then transfers and stops along the way are great. However, a great many students get picked up at a local station solving that mile problem without the need for further transfers.

Thus, while GO operates a valuable service, so does WUSA with Fedbus. Fedbus is faster, more direct, and only marginally more expensive. It balances out the comfort of the vehicle itself with less time waiting outside for the next bus to come.

	GO Transit	Number of GO Transit Transfers Required	Fedbus
YORK MILLS STATION	2h 39m	2	1h 45m
MCCOWAN STATION/ SCARBOROUGH CENTRE	2h 52m	2	2h 15m
RICHMOND HILL	3h 13m	2	1h 30m
CF MARKVILLE/ CENTENNIAL GO	2h 51m	2	1h 45m
CF MASONVILLE/ LONDON VIA	2h 43m	1	1h 30m
ALDERSHOT GO	3h 35m	2	1h 30m
JACKSON SQUARE	2h 49m	1	1h 15m

subdef

FRESHMAN

Cix

THIS IS A CAT, BUT YOU WON'T KNOW FOR SURE UNLESS YOU SUBSCRIBE TO mathNEWS++.

ONLY 35 HEADS OF CATTLE PER DOG YEAR

END OF TERM, BUT NOT THE END OF GRIDWORDS

gridCOMMENT *148.6*

It is honestly hard to believe that we've already made it to the end of the semester. I remember a few weeks ago making my first **gridWORD** for you all, crafting a brilliant (unbiased opinion obviously), **gridWORD** for you all. I can't believe we've come this far, solving some great (again, unbiased) **gridWORDS**!

I really hope you all enjoy doing them, and here's to many more in future terms! :)

Last week I had asked you what you: "What is a real game for true gamers?" There were some great answers to this question, some of them being:

- autotelic_individual: "Bullet chess"
- Aeschylus: "Recess game supremacy. Grounders. Manhunt. Four square. Cat and mouse. Oh, to be ten years old again…"

You all did great on this **gridWORD**, especially you autotelic_individual! I'm not the best at chess, but bullets must definitely make it more fun, I have to try that sometime 🤔

This **gridWORD** for this issue an oldie, back from 79.5, I am once again asking you to have fun with it (and also for your financial support). *[Editor's Note: This one's a cryptic, which we haven't had in a while, so go ham with it!]*

Until then, I'll see you all next issue everyone, and next term as well! Enjoy the break, and good luck with your exams! Keep up the **mathNEWS** grindset, and be sure to get your **mathNEWS**++ subscription >:)

Wink wonk

ACROSS

1. Raider shelled lieutenant (4)
3. Sombre farmer is death personified (4,6)
8. Dispatch Mrs. Simpson: rent from behind (8)
9. Record of letters from adolescent Ryan (5)
10. 4 points, united (4)
12. Truck congestion on highway (7)
14. Alluring legerdemain contains trap (8)
16. Upset in Versailles (3)
17. Lump is a pronounced bore (4)
19. Cow in tundra is bizarre; it has no right (5)
21. Shoo! Lousy cats (4)
23. Poorly marked route becomes a ditch? (3)
24. Somewhat mediocre mate deservedly incinerated (8)
26. Mach holds gamble with Scottish king (7)
29. Wean off again (4)
31. Without letup? I need limits (2,3)
32. Deep soprano notes heard (4,4)
33. Make one woozy with frantic excitation (10)
34. Vocally passes on colours (4)

DOWN

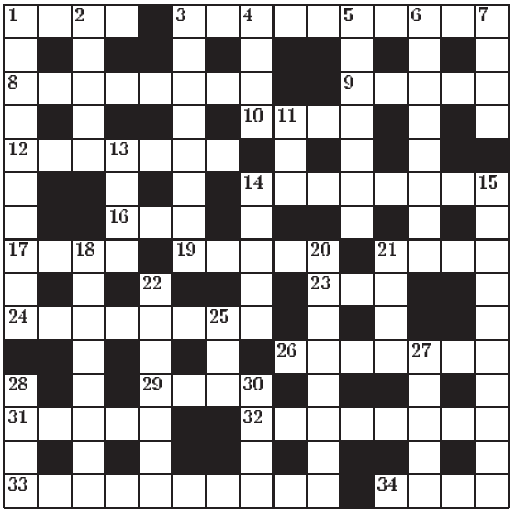
1. Opposing ecclesiastical tax mired in tomfoolery (10)
2. Delivered a spoken letter from Athens (5)
3. Wrote up rag on domain of famous cat (8)
4. Computer types NUCLEAR MISSILES: MISSING 100 (4)
5. Smoothing the transition from day to night (7)
6. My article's endless title is pitiful (8)
7. Rear speakers give energetic emissions (4)
11. Many years are topsy-turvy (3)
13. Exploit rotten fate (4)
14. Approaching death, after losing rib to pile of dirt (5)
15. Edited schematics for oral examinations (10)
18. Common visit (8)
20. Figure indefinite integral (8)
21. Drive winged creatures to the North (4)
22. Word for which this isn't a clue? (7)
25. Barely make out sound of shriek (3)
27. Yemen bombed foe (5)
28. Mischievous deity toned down for the audience (4)
30. Exclamation: "Oh! Aw, nuts" (4)

FEELING STUCK? LET gridWORD++ HELP YOU WITH EXTRA HINTS!

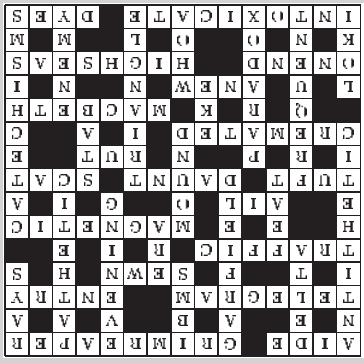
INCLUDED WITH mathNEWS++

•

ONLY ¤5ADCE8 PER QUAFON



LAST ISSUE'S gridSOLUTION



THIS ISSUE'S gridSOLUTION

lookAHEAD

MON APR 4	TUE APR 5	WED APR 6	THU APR 7	FRI APR 8	SAT APR 9	SUN APR 10
	Last day of classes	Pre-examination study day	Pre-examination study day Drop with WF ends	Final exams begin		

DISCOVER ALL THE BIRTHDAY PARTIES YOU'RE MISSING OUT ON WITH mathNEWS++.

ONLY ONE CAKE PER 18 TEST CHAMBERS

MON APR 11	TUE APR 12	WED APR 13	THU APR 14	FRI APR 15	SAT APR 16	SUN APR 17
				Good Friday No final examinations until Monday April 18		

PREDICT YOUR FUTURE WITH mathNEWS++.

ONLY 69 CRYSTAL BALLS PER KHZ^{-1}

THIS ISN'T FOR mathNEWS++ BY THE WAY. CRA RULES ONLY ALLOW US TO OFFER THAT IF YOUR YEARLY INCOME IS ≥$1 000 000 000 000.

YOUR NAME

MAILING ADDRESS

F W S 20YY F W S 20YY F W S 20YY

TERMS

CA **Canada**	7.50$ per term, 20$ per year
UM **United States**	10.00$ per term, 25$ per year
Overseas	15.00$ per term, 35$ per year

SAD IT'S OVER? SUBSCRIBE TODAY!

ISSN 0705-0410

UW'S BASTION OF ERUDITE THOUGHT SINCE 1973

mathNEWS is a normally fortnightly publication, funded by and responsible to the undergraduate math students of the University of Waterloo, as represented by the Mathematics Society of the University of Waterloo, hereafter referred to as MathSoc. **mathNEWS** is editorially independent of MathSoc. Content is the responsibility of the **mathNEWS** editors; however, any opinions expressed herein are those of the authors and not necessarily those of MathSoc or **mathNEWS**. Current and back issues of **mathNEWS** are available electronically via the World Wide Web at https://mathnews.uwaterloo.ca. Send your correspondence to: **mathNEWS**, MC3030, University of Waterloo, 200 University Ave. W., Waterloo, Ontario, Canada, N2L 3G1, or to userid mathnews@gmail.com on the Internet.

mathNEWS is overseen by the Board of Publications, an autonomous board of the Federation of Students, University of Waterloo, hereafter referred to as Feds. **mathNEWS** is editorially independent of Feds and the Board of Publications. **mathNEWS** has never been requested to withhold Improper Content as defined under Feds Policy 71.

Except where otherwise noted, this work is licensed under the Creative Commons Attribution-Noncommercial-No Derivative Works 2.5 Canada License. To view a copy of this licence, visit https://creativecommons.org/licenses/by-nc-nd/2.5/ca/ or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA. Terms may be renegotiated by contacting the **mathNEWS** Editorial Team.

mathNEWS++ is a fictional unregistered non-trademark of **mathNEWS**. April Fools'!