*math*NEWS

VOLUME 131 • ISSUE 1

FRIDAY, MAY 13TH, 2016

POKÉMON SUN AND MOON HYPE!

9 770705 041004

LOOK**AHEAD**

*math*NEWS

May 13	Rowlet turns its head 180 degrees to look at *math*NEWS
May 24	Popplio entertains writers
May 27	Litten stares indifferently at issue 2

MathSoc

May 13	Clubs meeting
May 13	Council meeting
May 19, 26	Games Nights
May 19	Clubs Budget meeting

University

May 13	Add period ends
May 20	Drop, no penalty period ends
May 21	Drop, penalty 1 period begins
May 23	Victoria Day

Miscellaneous

May 13	Friday the thirteenth

MAST**HEAD**

DOES ANYONE EVEN READ THIS?

Because we editors are incredibly lazy, unpaid and unappreciated workers have been hired to help us with several steps of our editing process, including proof-reading. Unfortunately, it just so happened that our workers are actually competent and know how to grammar good. Along with their complaints about not writing about actual "math" or "news" or having real "content" are those of our misuse of the passive voice. As shy, gentle creatures, aggression and directness are feared amongst mathies. since all of it is channeled through our mathematics, our primary form of communication. Natural language, being secondary (or tertiary), suffers as a result. However, our faults are not unknown to us, so we asked our writers: "What things in life should not be passive?"

Zethar("I feel that I am of the very reasonable opinion that life shouldn't be passive."); Joseph Plumber("Life, theatre, and choosing the next President of the United States"); me("Aggression"); Diminutive Rex("Voice"); Element("The spread of communism"); DeadPool("Pegging"); Pockets("dance"); Beyond Meta("Pacifism"); waldo@<3.LE-GASP.ca("Not sure. I just want to pass my classes");

bunniED("My enthusiasm for Pokémon Sun and Moon. Did you see the trailer video? I never knew I needed a cute orb-like owl with a leafy bowtie. So cute! Also, the graphics have improved so much from Pokémon X and Y; the proportions are all different! I didn't think more improvement was possible for a Pokemon game.").

FORT MCMURRAY

On May 1st, 2016, a wildfire started southwest of Fort McMurray, Alberta. It grew out of control very quickly. In the past two weeks, it has moved into the city, which issued a mandatory evacuation order on May 4th.

Thankfully, most civilians were evacuated safely. However, huge sections of the town were destroyed. Currently the town is totally empty. This firestorm might become the most costly natural disaster in Canadian history, with an estimated cost of up to 5 billion dollars.

However, is this really a natural disaster? This year has been the hottest year on record. Northern Alberta experienced record temperatures of up to 30 degrees and unusually dry weather. (I am not suggesting anything; I am just saying you should consider the increasing temperature all over the globe and its correlation with the burning of fossil fuels.)

Currently, the Government of Alberta and the Government of Canada will match any donation you make. For example, if you donate $10, you actually donate $30. One method of donating is via the Canadian Red Cross: `donate.redcross.ca`, but there are other channels.

me

IMPROV TV LIVE TAPINGS

Next week at KWLT, Rogers is taping the live shows of the improv troupe Theatre on the Edge. Come be part of a live studio audience! Take the opportunity to shape the show with your suggestions! Be on TV!

Shows are Monday–Friday 7PM and 9PM at 9 Princess St. E., Waterloo. Pay what you can (essentially free). Laugh, learn, and tell all your friends and all your enemies too. We desperately need bums for seats and may literally fill them with bums if need be.

Beyond Meta

FIXED POINT THEOREM

If you scrunch up a pizza and place it on top of another pizza, then there is a point on the scrunched up pizza that matches the non-scrunched up pizza.

IDRIS TUTORIAL V2: THE IDRISSENING

CONTINUED FROM V130I5: AN IDRIS TUTORIAL

Last term, I talked about cool types like `Vect`, but this issue, I will actually tell you how to define them!

You might already be used to the following kind of syntax for defining types:[n]

```
> data List a = Nil
> | (::) a (List a)
```

In Idris, we use a different syntax[n] for this definition:

```
> data List : Type -> Type where
> Nil : List a
> (::) : a -> List a -> List a
```

We say "`List : Type -> Type`" because `List` is a *function* that takes a type and returns a type[n]. So we can say:

```
> map List [Nat, Bool, List Nat]
Result: [List Nat, List Bool, List (List Nat)]
```

Now that we have specified what `List` is, we define the constructors by just giving their name and their type.[n] For instance, `Nil` has the type `List a`[n].

Now, we can use this new freedom to add some nice new fields to our lists:

```
> data Vect : Nat -> Type -> Type where
> Nil : Vect Z a
> (::) : a -> Vect k a -> Vect (S k) a
```

`Z` is just the number zero, and `S` is the successor function.[n]

Some interesting corollary of this that sometimes, you can not use some particular constructor to create something of some type. For example, you can't use `Nil` to get some object of type `Vect 5 Bool`. You can even make this more extreme!

```
> data VeryExclusiveType : Nat -> Type where
> FortyTwo : VeryExclusiveType 42
> FourTwenty : VeryExclusiveType 420
```

While this definition gives you an obvious way to construct an object of type `VeryExclusiveType 42`, it doesn't seem obvious how to construct something of type `VeryExclusiveType 69`. In fact, it is impossible.[n]

There are other mischievous things that you can do when defining data types to prevent the user from creating objects, like for example this:

```
> data CoolBool : Bool -> Type where
> TrueIsCool : CoolBool True
> FalseIsNot : VeryExclusiveType 13 -> CoolBool False
```

So why do we care about all those types that have no values? The answer is the *Curry Howard isomorphism*, which is just the fancy way of saying that we can interpret *types* as *propositions*, and *programs* as *proofs*.

So how does this work? Suppose we define the following two types:

```
> data Void : Type where
>
> data Something : Type where
> C : Something
```

We notice that you can't construct something of type `Void`, but you can construct something of type `Something`. Thus, we say that `Void` is *uninhabited*, and `Something` is *inhabited*.

So if we interpret inhabited types as being *true* and uninhabited types as being *false*, we can treat types like propositions. If you construct an object of some type, you demonstrate that the type has to be inhabited, so objects of a type can be interpreted as proofs!

But now, if we combine types, for instance by forming the type of functions from some type to another type, how do we interpret the resulting type? Let's try answering that by checking if some function type is inhabited:

```
> voidtovoid : Void -> Void
> voidtovoid x = x
>
> voidtosomething : Void -> Something
> voidtosomething x = C
>
> somethingtosomething : Something -> Something
> somethingtosomething x = x
```

So now we showed that those three function types are inhabited, but what about `Something -> Void`? As it turns out, in Idris, while you *can* get something from nothing, there is no way of getting nothing from something.[n]

So how should we interpret function types? As implication, of course, since `A -> B` is inhabited if and only if "if A is inhabited, then B is inhabited" is true.

So types are propositions, what is something like `VeryExclusiveType`? It is not a type (i. e. something of type `Type`), but rather something of type `Nat -> Type`. So it makes sense to think of it as a *predicate*.

So now we used data types to construct a predicate `P` such that `P(42)` and `P(420)` are true and `P(x)` is false for every other `x`. Can we build more useful predicates? Yes, we can! For example,

```
> data Even : Nat -> Type where
> ZeroIsEven : Even 0
> PlusTwo : Even k -> Even (S (S k))
```

`Even n`, is, unsurprisingly, inhabited if and only if `n` is even.[n]

We can even build predicates that take two arguments!

```
> data Leq : Nat -> Nat -> Type where
> ZeroIsSmallest : Leq 0 k
> SPreservesOrder : Leq a b -> Leq (S a) (S b)
```

This is pretty neat[n], but how does it help us when writing programs? Now, we can write functions that require their arguments to satisfy certain conditions. For example, you could write a function that divides a number by two that only accepts even numbers (and, in fact, only accepts a number if, at compile time, you can *prove* that it's even):

```
> div2 : (a : Nat) -> Even a -> Nat
> div2 Z ZeroIsEven = Z
> div2 (S (S x)) (PlusTwo proof_for_x) = S (div2 x proof_for_x)
```

So now, if you always pass around certificates that all the numbers that should be even are even, your programs will never crash! Yayyy!

If you find that example pretty silly, you can have another one: We can define some predicate for "`x` is in the list `l`":

```
> data Elem : a -> List a -> Type where
> Here : Elem (x :: xs)
> There : Elem x xs -> Elem x (y :: xs)
```

Then, we can define functions like `deleteFirst : (x : a) -> (l : List a) -> Elem x l -> List a`.

If you find this example pretty silly as well, I guess I can't really help you. But whatever your opinion on this thing is, I hope you enjoyed my Idris tutorial, because it is probably over riiiight now!

Daaaaaank

[n] `(::)` is just the "cons" operator in Idris. I guess they just wanted to confuse Haskell programmers by switching "`:`" and "`::`".

Also, we are going to define the operator for several types. The compiler will figure out which definition we mean when we say `a :: b`.[n]

[n] At least, I hope it will.

[n] The syntax I just showed is allowed as well, but you can't use it for all the fancy shit that I will introduce here.

[n] whoahhh

[n] If you ask the compiler for a constructor that has a "non-constructor-y" type[n], you will just get an error.[n]

[n] Like this:

```
> data CoolType : Type where
> MagicallyGetSomeValueForEveryType : (a : Type) -> a
```

[n] I am anticipating that you are kinda confused how the compiler figures out the right constructor for your type, so I will actually try to stop handwaving it, and start to feet-wave it or something:

If I define a data type T, then its type has to be of the form α_1 `->` α_2 `-> ... ->` α_n `-> Type`.[n]

Every constructor C_i has to have a type of the form a_1 `->` a_2 `-> ... ->` a_m `-> T` β_1 `...` β_n, where every β_k is an expression of type α_k.

As the type tells you, if every x_k is an expression of type a_k, then C_i x_1 `...` x_m is an expression of type `T` β_1 `...` β_n.

More interestingly, if every p_k is a pattern that matches something of type a_k, then C_i p_1 `...` p_m is a pattern that matches something of type `T` β_1 `...` β_n.

Since creating them and pattern matching them is the only thing that you can do with objects of the types you can define, my explanation is now officially done, and any remaining confusion is your problem.

[n] Actually I am still lying to you: You can, of course, use dependent product types[n]!

So the type of `T` can actually be of the form `(`γ_1 `:` α_1`) -> ... -> (`γ_n `:` α_n`) -> Type`, where all the γ_k are variable names, and where α_k is an expression that may use the variables γ_1 to γ_{k-1}.

[n] "Dependent product types" is just the name for types of the form `(x : A) -> B`.

[n] Actually, it has the type `{a : Type} -> List a`, so the compiler can try to figure out what `a` is depending on the context.

[n] `Z` and `S` happen to be the two constructors of the data type `Nat`, but that's not as important as it might seem. We might as well have said "`Nil : Vect (3 * (0 + 0)) a`".

[n] Ohh, actually, I am blatantly lying. There is the function `believe_me : a -> b`[n] that converts an object of any type into an object of any other type, if one day, you become lazy and just want the compiler to, well, *believe* you that the program you are writing is correct.

[n] Or, if you are still confused by implicit arguments, `believe_me : {a : Type} -> {b : Type} -> a -> b`.

[n] If you were planning to write me an angry letter pointing out the existence of `believe_me`, you get a cookie for being very smart.

[n] Exercise: Write a data type `Odd : Nat -> Type`!

[n] Another neat thing: You might be wondering where, after I introduced the `->`-operator, all the other logical operators went. The answer is that you can think of a logical operator as a *predicate*, that, instead of taking a number or whatever, takes in *two truth values*, that is, types. For example:

```
> data And : Type -> Type -> Type where
> Both : a -> b -> And a b
```

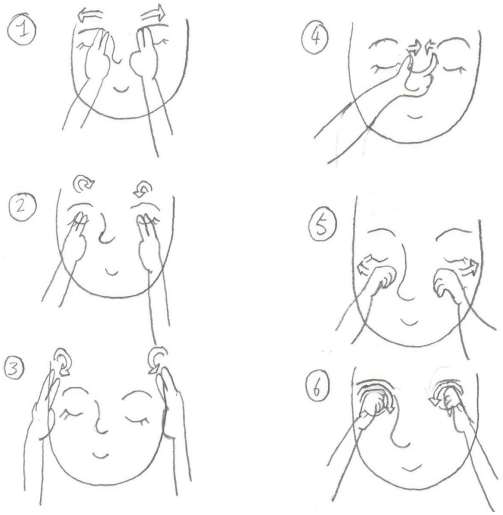
As an exercise, you can think about how you would construct `Or : Type -> Type -> Type`.[n]

[n] By the way, the type that corresponds to "not `A`" is just `A -> Void`.

EYEBROW EXERCISES

EYEBROW EXERCISES ARE THE BEST THING EVER!

There is a set of eyebrow exercises that my elementary school taught us to do to protect our eyes from damage due to studying too hard. You should do each of these exercises 8 times for a total of 48 exercises. Do it every time your eyes get tired from studying too hard!



1. First, put your index and middle fingers together on both hands.
2. Push both sets of fingers on the eyebrows, near the nose, and pull them to the outside, gently rolling over both of your eyebrows.
3. Then, put your index and middle fingers on your (closed) eyes, and rub them in a circular motion.
4. Then place this set of fingers on your temples and rub those.
5. Then, take the thumb and index finger of your favourite hand, and grab the parts of your eyes that are closest to the nose. You should feel two squishy little bits; squish those.
6. Then take the index and middle fingers of both hands again, and fold them up as pictured.
7. Then, use them to rub your cheekbones. After that, you can also place them over your eyes like little goggles, and then massage the browbones.

In conclusion, eyebrow exercises should be the top sport in any Olympics. Also, one should establish an International Olympiad in Eyebrow Exercises, where students from all over the world come together to celebrate this glorious sport.

kartoffel & spargel

N THINGS THAT PMC PRESIDENT WILSON WOULD PROBABLY HAVE WANTED ME TO INCLUDE IN MY IDRIS ARTICLE

1. When we write `Type`, the compiler magically attaches some number to it. Why do we need that? Because, for some reason, if we had `X : X` for some type `X`, we would run into Russel's Paradox somehow.

So we can write `Type : Type`, but Idris translates this to $\texttt{Type}_{\texttt{k}}$: $\texttt{Type}_{\texttt{k+1}}$ for some number `k`, so we are magically saved from paradoxes.

The only time when you need to worry about this is when you get the ominous error message "Universe Inconsistency", which means that the Idris compiler failed to invent good numbers to attach to all the `Type`s.

I couldn't even think of an example to demonstrate that, but President Wilson probably wanted you to know it anyways, and a lowly human like me has no grounds to oppose a godly creature like Him.

2. Dependent product types (`(x : A) -> B`) are really cool when you interpret them as propositions. B is a formula that may involve the name `x`, so if you have a function of a type `(x : A) -> B`, you know that *for every* `x`, `B` is inhabited.

For example, `(a : Type) -> a -> a` is the type corresponding to the proposition "For every a, if a is true, then a is true", which is proven by the identity function.

3. The evil twin of the dependent product type is the dependent pair type `(x : A ** B)`, where `B` is again a formula that may involve `x`. It is the type of all pairs whose first element has the type `A`, and the type of the second element depends on the first element. When Curry-Howarded, this corresponds to "there is some `x` of type `A` such that `B`".

4. Apparently, you can make data type definitions depend on recursive functions that, in turn, depend on the data type, like in this code snippet that I stole from Reddit:

```
  > mutual
  > data U : Type where
  > Top : U
  > Bot : U
  > Sum : (t : U) -> (inj t -> U) -> U
  > Prod : (t : U) -> (inj t -> U) ->
U
  >
  > inj : U -> Type
  > inj Top = ()
  > inj Bot = Void
  > inj (Sum c x) = (v : inj c ** inj
(x v))
  > inj (Prod c x) = (v : inj c) ->
inj (x v)
```

This pattern is called "Induction-Recursion" and is apparently important for something or whatever.

Dank

FORT MCMURRAY FIRE LINKED TO CANADA GEESE

New information from Fort McMurray has just confirmed that wild geese are indeed responsible for the raging forest fires.

Fort McMurray fire chief Darby Allen reported, "At first we just chalked [the fire] up to the abnormally hot temperatures, but after we saw the third flaming goose, well, we started to wonder."

Further investigation found an abnormally high quantity of strange looking geese in the area, and locals recall seeing many of them around just before the fires started. The geese were generally described as red-eyed, horned, fiery, and extremely vicious.

"Just like regular geese, really," commented a recent evacuee.

When asked about special measures he intends to employ against the emerging goose threat, Chief Allen mentioned that the force has been experimenting with deploying roving bands of hardened UW Students: "Some of those students have been through more shit than my most experienced men—and I mean that literally."

aPlayerofGames

DONUT COROLLARY

If you allow self-intersection, you can turn a donut inside-out. Donuts are fricked up.

DON'T MAKE OTHER PEOPLE'S MISTAKES

MAKE YOUR OWN

One* time I made a mistake. I told a story that made people incredibly uncomfortable. Upon realizing it, I was mortified, thinking all sorts of "productive" thoughts, like "no normal person would ever have made this mistake" and that people would dislike me for it.

Later, when I was not horribly sleep-deprived, I realized that the metric I was using wasn't great. I am not like other people and I should not expect to repeat the same errors as them. Mistakes are great in that they come in a variety of forms. If they were easy to identify, we would make a lot less of them. I will make missteps that other people would not, but I will also gain experiences that others might not. Life is process of discovery, and are mistakes are a part of that. The only way to avoid them is to stop trying new things and never go out of our established boundaries, which I would say is the greatest mistake of all.

There will always be new mistakes to make. The question is: will you acknowledge them and try to improve, or ignore it? Realizing that you have made a mistake is not an indication of your character, but rather an opportunity to learn and do better.

Beyond Meta

*This may be a slight understatement**

**That was also an understatement

BREAKING TRISCUIT NEWS

A PUBLIC SERVICE ANNOUNCEMENT

Two Wednesdays ago, I was in my local grocery store, lamenting the fact that Triscuits cost way more now than four years ago, when I came across a flavour of Triscuit I had never seen before! BEHOLD:

These are "Smoked Gouda" Triscuits. If I am not

mistaken, this is the second Triscuit flavour to feature cheese, behind "Parmesan Garlic", a Thin Crisp flavour. Hence, it is the 'main' Triscuit flavour, i.e., not a Thin Crisp flavour, to feature cheese.

And wow, are they good. They have a strong cheesy taste, with a subtle smokey hint, and the combination of this with the standard Triscuit cracker flavour makes for a fantastic snack. Unlike other Triscuit flavours, it is not necessary to adorn these Triscuits with cheese, allowing you to top them with meats or veggies or in general, antipasto. However, the lack of any herbs or spices makes these Triscuits a bit less exotic than otherwise.

Look for this new flavour, apparently in stores now! This has been Breaking Triscuit News. (There really should be a pun on 'crackers' here. Sorry, Beyond Meta.) Happy munching!

Scythe Marshall

SKETCHYCOCKTAILS.WEBSITE

So it is Friday night, and the ceaseless tide of assignments, midterms, interviews, and reddit shitposts has got you down. You go to the ol' booze closet, but it has been weeks since you had the time to go to the LCBO, so all that you have left is half a handle of rum, some weird craft beers your roommate gave you, a couple of those parasitic freebie bottles of fruit vodka and a full litre of melon liqueur. Your fridge has a couple cans of orange crush left over from the last pizza run, some questionably expired lemonade, and litres upon litres of chocolate milk. What do you do?

`SketchyCocktails.website` is here to help. Just list everything that you've got, and it will provide suggestions of what to mix up. Maybe you should try mixing the vodka, melon, and lemonade. Perhaps the beer, orange crush, and a dash of chocolate milk is more up your alley. Whatever you choose, you can then give it a rating to let the community know if it is worth a taste.

Now, we need some alpha testers to load up our website with esoteric combinations and rate them. So, if you are interested in a co-op job that involves you mixing up really weird and probably terrible drinks and pouring them in your mouth, consider applying to us. We are not going to pay you. We will probably not even pay for your materials. But, and we are going to be honest here, this does not even put us in the bottom quartile of jobs available on Jobmine.

`SketchyCocktails.website`: whatever life choices led you to us, we will help you forget them.

Diminutive Rex

N THINGS I'M DOING INSTEAD OF FINISHING MY WORK TERM REPORT

1. Facebook
2. Watching Game of Thrones
3. Reddit

3.5 Hyping out over Game of Thrones fan theories

4. Watching Silicon Valley

4.5 Not reading Silicon Valley fan theories because the show hits too close to home

5. Reading over (but not doing!) other assignments
6. Getting LaTeX setup
7. Coming up with the perfect report title
8. Twitter
9. Calculating how much time I have left before the LEARN dropbox closes
10. Calculating how many words per minute I need to write to get it submitted in time
11. Writing a pros and cons list to determine if I should defer writing a work term report to my next co-op term
12. Planning what pizza flavours to request at *math***NEWS** night

n. Writing this article

HatOfChocolate

MINORITIES DISCOVERED NOT TO BE HOMOGENOUS HIVE MINDS

Researchers discovered last week that members of minority groups do not, in fact, all share the same opinions. They also found that one's membership in a group does not remove from the fact that they are also individuals. This suggests you should not assume that what one member says represents the views of the whole group. Despite the usefulness of knowing common features of a group, experts suggest that the experiences of a group are a lot more diverse.

Researchers caution against taking this new finding as reason to not ask members of minorities about their views on an issue. They instead recommend asking multiple members, as one person is too small of a sample set to gain a good understanding of an issue.

Beyond Meta

LOGISTICS

Imagine that you haven't had a good get-together with your friends in a while and you want to organize a sushi meal. You ask your friends when they are free in the next week. One friend tells you immediately that they are busy on Tuesdays and Thursday due to night classes. Another friend tells you that they work all day on Saturday and Sunday and needs to run a club meeting Monday evening. The third friend said they're free whenever, and the fourth friend didn't get back to you.

OK, so you go. Let's choose the sushi restaurant. Apparently someone got food poisoning at Sushi 99 last time they went, and you read a review that 168 Sushi serves rubbery imitation pink bricks masquerading as salmon. You want to go to Kinkaku's in Kitchener, but they are far and don't take reservations. One friend says that they want to go to an all-you-can-eat place since they have a large appetite and you really don't want to imagine the logistics of an *à la carte* sushi experience, especially since you aren't willing to touch sashimi despite that your friends like that stuff.

Eventually, you decide on a place and a time on Wednesday. Then your third friend casually mentions that they have a dentist's appointment on that Wednesday so you call back to change it to Friday, but then they say that they're full on Friday so you change it to next Wednesday, and tell all your friends.

Finally, your fourth friend gets back to you and says that he's sort of busy this term and is only free on the weekends. You call back and reschedule the reservation a second time, but then your first friend says that they are going to out of town that weekend. Now you're wondering if it's worth rescheduling this dinner for four weeks in advance, during exams, when everyone's schedule is probably going to be different…

Now imagine if this is an event that is supposed to be regularly occurring, like a tabletop roleplaying game.

Zethar

HAIRY BALL THEOREM

OF COWLICKS AND CYCLONES

When one attempts to comb a hairy ball flat, there will always be at least one tuft of hair at a point on the ball.

The Hairy Ball Theorem can be applied to explain the behaviour of cyclones.

N THINGS FORBES THINKS ARE APPROPRIATE BUSINESS EMAIL SIGN-OFFS

- High five from down low
- Peace dude
- A smiling face is miles more attractive than just a pretty one.
- XOXO
- Rgds
- Snuggles
- Pardon my monkey thumbs
- Now go do that voodoo that you do so well!
- [:-)
- Have a wonderful bountiful lustful day

Source: 89 Ways To Sign Off On An Email by Susan Adams

Have a wonderful bountiful lustful day,

Diminutive Rex

OVERWATCH: UNCOVERED

I have really enjoyed playing Overwatch during the open beta that ended last Tuesday. Apparently, a number of people have been enjoying Overwatch in a different way: according to an article on Venture Beat, searches for Overwatch on PornHub jumped 817% during the open beta.

Now, my poison of choice is fanfiction. I decided I was going to scour the web for the worst Overwatch fanfiction and give it to you. Except… I couldn't find any!

I am no slouch in the world of weird fanfiction. I have read *Flowers In A Box*, I have seen Hogwarts fuck the giant squid, and I have read all of *My Immortal*. The fact is, the worst thing you can say about a fanfiction is that

it is mediocre. There is nothing interesting about the Overwatch media that has been written so far. Tracer and Widowmaker insult each other a bit and then have sex. After reading the previous sentence, you have the gist of 50+% of the Overwatch fandom. Sure, there are a couple weird digressions, like Roadhog getting it on with Junkrat, or the story entitled "Fucking an Icicle stuck inside a Mech trying to kill a Spider at the Speed of Fast", which is 1500 words of bizarre lesbian sex—but they are few and far between.

I am honestly disappointed. I was hoping to let you in on a new weird world of kinky insanity, and instead I have been left empty-handed and unsatisfied. …say, that's a good turn of phrase. I could work that into a story about Reaper, after he throws away his guns… yeah, and he can run into Lucio in one of the mining tunnels…

You know what? I just had an idea. I'll come back to this article later.

Diminutive Rex

PROFESSOR ANNOUCES SURPRISE FINAL

With less than two weeks remaining before the final exam, a professor decided he didn't really like the exam date, and instead opted to have an in-class exam on the last day of classes and a take-home final. The reaction in the classroom was one of collective horror, as an exam for which they previously had a month to study was moved to a week where all major assessments are due. When students asked where they would find time to study for the exam, the professor replied that they had the entire semester to study.

The students' only consolation was that their prospects for succeeding in the exam looked better than the professor's prospects for course evaluations after such a decision.

BeyondMeta

BLURRINESS OF SELF

PART 1

Firgaz stood atop the observatory in the midnight gale, light cloud cover above.

"Ah, how nice it is to see so many stars at dark again…" she muttered.

She then walked towards the telescope stand. Hands shaking from the cold, she held on to the stand and looked through. At first, she saw blurriness.

Oh, I forgot. I have to calibrate the telescope a bit. I haven't done this in so long…

She turned to the complex mechanism on the side of the telescope and tried to recall the procedure required.

If only I studied mechanics instead… but then I wouldn't be up here at all. I feel like I remember, but I actually don't…

Struggling with the machinery, she found settings that worked after several minutes of fiddling. She then let out a sigh and set her eye in the telescope again.

Her first point of interest lay quite obviously in the sky, but she wanted a closer look at it. The Star of Ermahe. Firgaz had heard that Ermahe was scheduled to put on a rather dazzling display today, but when she found it, she was quite disappointed. Sure, it looked different, but neither did it look brilliant. This time, there was merely a few more glowing lights than usual.

I waited the entire year for this? But, well, weather forecasters are awfully inaccurate, so I suppose stars are as well. … Now, what was it again? What to see next?

As she was, struggling to recall, she retrieved her notebook, only to discover that she had forgotten to write anything regarding this night. How could she be so careless?

Ah, well, I'll just look around. I haven't stargazed in so long and won't be able to for a while.

Scanning the skies, she eventually found her favourite constellation: The Irme Triplets. Only… something was missing.

The fourth star? I can't see it. Must have disappeared. Well, the heavens change daily.

Somehow, despite this new development, Firgaz felt no better about this whole stargazing thing. She only scanned the skies a little more before abandoning her post.

"Oh… I guess I've just wasted my time here. All I've figured out is that I don't like astronomy anymore. Just some childish dream."

She packed her bags and walked down the tower's stairs. For someone who needs it more. No one would use the tower for a little while after that, but it was the intent that mattered.

Firgaz stumbled through the campus for some time before finding the train station. She had never bothered to memorize the layout of the university. That was not totally inexcusable, due to its vastness and unintuitive design, but for a fifth-year student it was rather odd. She

arrived home quite late and exhausted, with her eyes blurry. Her partner Helse was, as usual, awake and sorting some documents.

"Well, that didn't work. I couldn't recapture the enjoyment I used to have for astronomy," Firgaz said.

"Oh, but you never enjoyed astronomy that much. I don't even know what drove you to go out today anyway. You hardly ever go out."

"I- well, I came to see the Star of Ermahe, but that was kind of a disappointment. I just heard it would be really pretty… and, well, it wasn't."

"You and pretty things. I've never understood that."

"Er… there was something else too. I don't remember what it was. All I know was that it was important."

"You and your forgetfulness too."

"Ah, it probably doesn't actually matter. We should get to sleep."

"We still have an assignment due tomorrow. Theory of Magical Sealing. Don't tell me you've forgotten that too!" Helse replied.

"I didn't actually forget. I just thought you would–"

"You know I wouldn't do all of it for you."

"Oh, all right."

"But nevermind. I've already done most of the work anyway. You'll just need to fill in a few loose ends."

"But-"

"I said most, not all."

"All right. But I'll do it in the morning. I'm so tired…"

"Ah, okay… I'm pretty tired too, actually. Let's sleep."

namelessRegent

PROF**QUOTES**

"This is actually relevant to the course. I swear it's relevant."

Avery, CS 349

"Don't post your code in GitHub! After the course is done, print it, I don't care. Just don't post it on GitHub!"

Avery, CS 349

"This is the time when I'm not doing anything other than twiddling my thumbs."

Avery, CS 349

"I like to say I can't draw, but I can at least draw better than that."

Avery, CS 349

"It opens a window. Ooh. You're supposed to be impressed right now."

Avery, CS 349

"I'm not sure why you do this except that it's really cool."

Avery, CS 349

"This is a bunch of stuff you don't have to read."

Avery, CS 349

"Without the digital display, you can't open the microwave when there is one second left."

Avery, CS 349

"Teaching is all about contrived examples."

Avery, CS 349

"Don't plagiarize. Except for Assignment 3, where you can plagiarize as much as you want off your partner."

Daudjee, CS 454

"No, I didn't misspell 'Windoze'."

Daudjee, CS 454

"You're stealing my thunder. I had a nice segue into that."

Weddell, CS 348

[*During a 2:30PM class.*] "If you find that you sleep in too late, you can attend the section at 1PM. I think there's a temporal issue here."

Weddell, CS 348

"I really like it when you ask questions. [*Students raise hands.*] Not right away."

Weddell, CS 348

"I bet none of you will remember to do that and will all go to the same room. You'll then learn in real time about the inefficiency of linear search."

Weddell, CS 348

"I worked for a company called Nortel. Who's heard of Nortel? I want to point out that it was not my fault."

Weddell, CS 348

"Which room do you write your midterm? I suggest flipping a coin twice."

Weddell, CS 348

"I'm sure one of you is worried about chaos, which is what usually happens when a professor is entirely responsible."

Weddell, CS 348

"You're not allowed to be bored in this course. If you find yourself to be bored, come see me and I'll give you stuff to do."

Weddell, CS 348

"You have to go back and change your experience of [CS] 245 so that you loved it."

Weddell, CS 348

"When I say 'theoretical', does that mean my course evaluation will be down?"

Weddell, CS 348

"Unfortunately, assignment 1 is all in English."

Weddell, CS 348

GRID**COMMENT**

Hey everyone. We are missing a ConvolutED this term and thus at the last minute for this issue I was asked to make a puzzle. I did, so this means that I have officially become the gridmaster after way too long in writing for this publication. Since I am absolute garbage when it comes to crosswords, this puzzle isn't one; it is instead text encoded in a well-known cipher pilfered from crazy conspiracy nuts who don't understand what is considered cryptographically secure. It is up to you, dear reader, to decipher this message and relay it to us via submission, either electronically to mathnews@gmail.com or via a physical piece of paper dropped into our ever-hungry **BLACK BOX**, conveniently situated outside the Comfy Lounge.

Don't forget to sign your work so that we may publish the results. In the event of more than one submission, those eligible (using the metric of closest to the cleartext) shall square off and answer the *grid***QUESTION** with the person giving my favorite answer being able to claim a prize from the *math***NEWS** editors. This issue's *grid***QUESTION** is "What is the best conspiracy theory?"

Yours in the shadows,

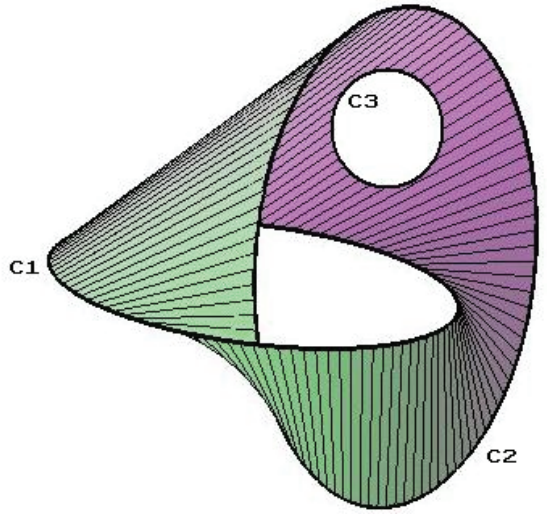
Zethar

FACTS ABOUT PANTS

- They have three holes.
- There is no distinction between leg holes and waist holes.
- They can be sewn together leg-to-leg or leg-to-waist.
- They consist of two hexagonal fundamental polygons stitched together at every other side.

FIGURE: A PAIR OF PANTS

FROM WIKIPEDIA




```
tsmokhgahipvyrpglbkkogmwvokfnymfkrkagwmpdntsnnnevyztupobvomklwprdtcfcpnlgrmpvtkflcfgpkgfgvutnfdtdsvozpnaxbuttuowmfprdtpamknryryi
ygpasygayzpgyrkuxwpyfxvtlcpflbttktuskffgbocghlaozpnmfupwdtabsrobnbucoopvyvfscsgapxiefudqpglbskfuycfrnbnfubxctbhokynlsynekxkfdtnf
nvfxgwkxkcnafffuybfuycfrmffyhcusawnfdtmtyfprnblkendtdsvozpnaxbdvvxmksqyrowgxvognzoktoblnutnwdtboxbiwtbhuxgmwvokflcttxtobllxtyucr
nfdfnylhcwyghvnfnvfxiqlkwpkfnftohfnimpfuwagxpasyebixgrgozpwlnbgahipvyrpglbttktvppggapigpsbtxpryxyrpdhukueesygapxiefudqpglbkftbsw
pbqszptugafrmwvokfidlwlknrkwdtnlqsgnaalfgafrlwzvnaxbygowfrechskxkcpdtagavrkyqsgnlwewswfvsakyqvxbkhgahipvyrpglbkkogmwvokfztnfdtne
oezazxnfnvfxiokoebxtyucrztktgagranfugwgayxgafrtbyraalfnbpbcscwnwdtklkyfrpktufvmohanlkndtnfnvfxgwlunndtcrmlxwpxucfsublwpgobonmwvo
kfkcdslkqpqsnnlyiyzpkfnfdtdsvozpnaxbgxfyfvdvvbsqyrnbmfcxmpgfnfnvfxzopfuqlwdoidlyfylwpbfrkkzoanvveegagaowgycwkskkiohvtusqpvnbnffu
pqprgafrubfxibpcswlwewkopgfifgkkiwowfxnbfyfudpgaownfpggrmgtquqhuyghztbgvlbktsqsdlwlfdtfrzohlowakfxzoktutqslwtqyrganlblekekekekek
```

ISSN 0705—0410

FOUNDED 1973

***math*NEWS** is normally a 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. ***math*NEWS** is editorially independent of MathSoc. Content is the responsibility of the ***math*NEWS** editors; however, any opinions expressed herein are those of the authors and not necessarily those of MathSoc or ***math*NEWS**. Current and back issues of ***math*NEWS** will eventually be available electronically via the World Wide Web at `http://www.mathNEWS.uwaterloo.ca/`. Send your correspondence to: ***math*NEWS**, MC3030, University of Waterloo, 200 University Ave. W., Waterloo, Ontario, Canada, N2L 3G1 or to `mathNEWS@gmail.com` on the Internet.

EDITORIAL STAFF

Katherine Tu (Sun)

Zishen Qu (Moon)

SO SHINY!

IMAGES FROM SEREBII.NET