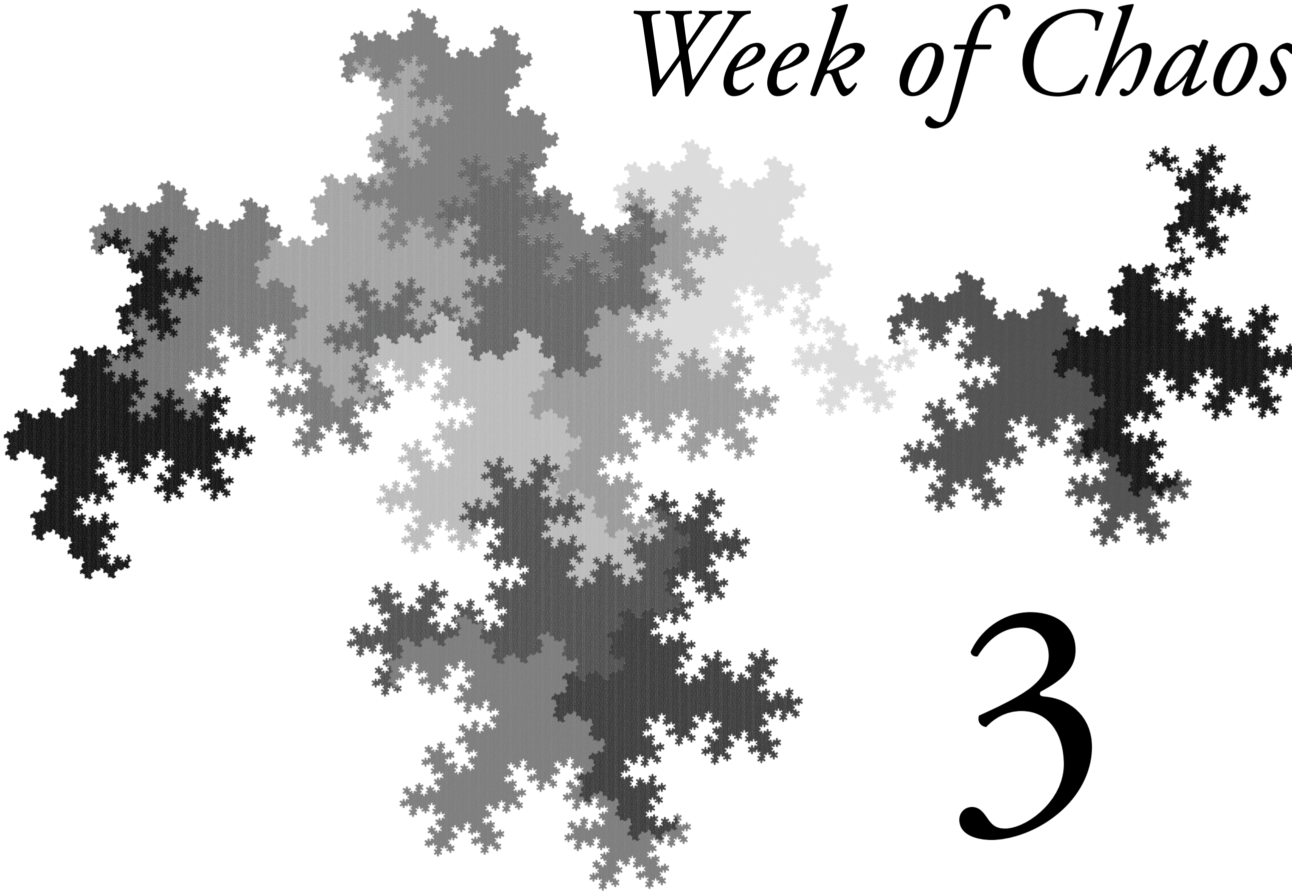
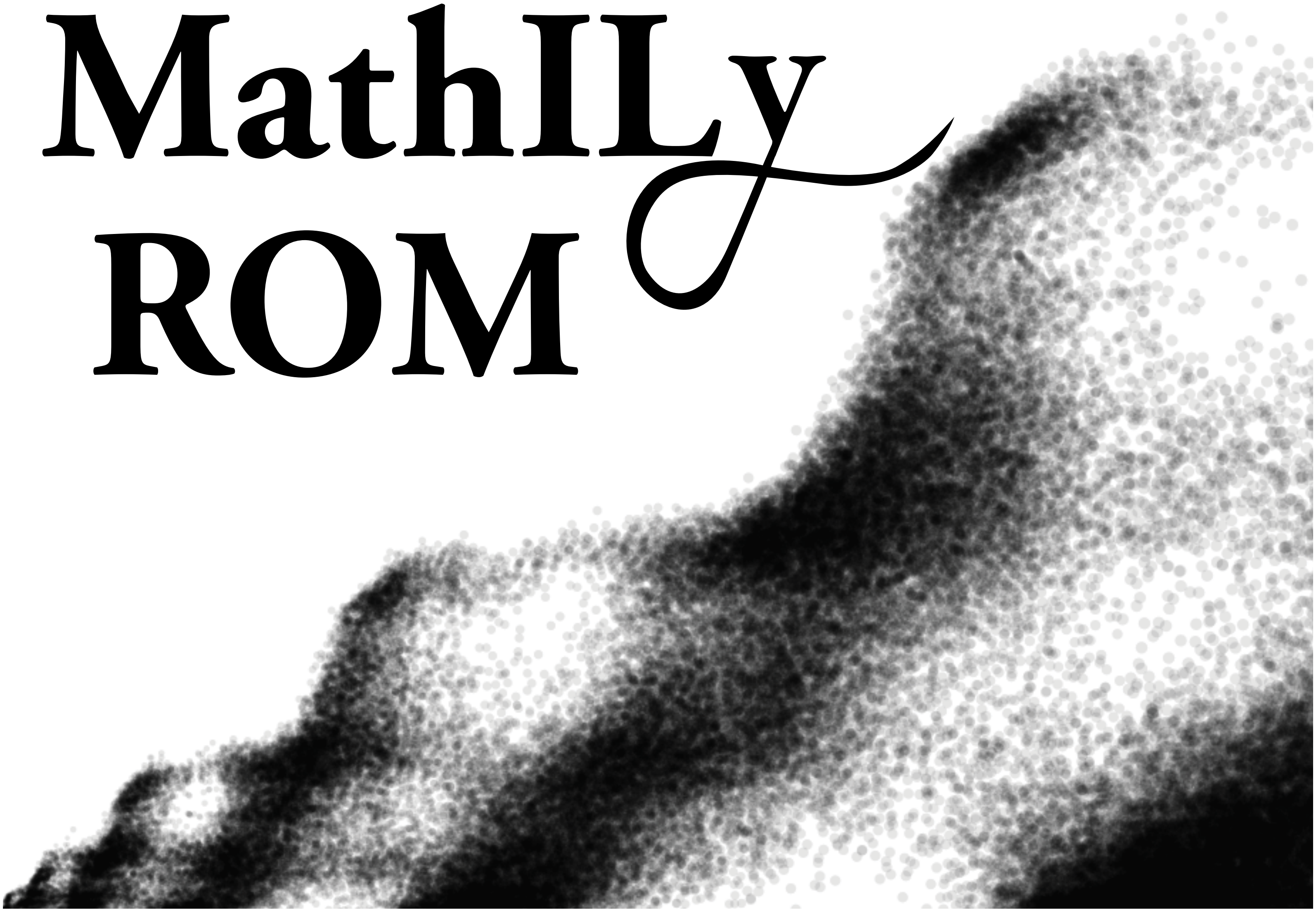
Week of Chaos!



3

MathILy & ROM



Contents

Also please play this fun game: https://bit.ly/2NxM2mf

Editors: Alex, Justin, Lillian

Cover by: Annika

1 Class Summaries

Big Finite Diff (Tom)

by Jon S

Welcome to the big world of Big Finite Diff(erences), where we consider different ways of computing sums and discuss differences of terms in sequences. We learned some new notation, falling factorials (denoted as x^k) defined as $(x!)/(k!)$. We also learned of a rip-off Pascal's Triangle, the Stirling Numbers! It turns out that Big Finite Diff has striking similarities to a certain C-word, but finding that will be left as an exercise to the reader.

Non-Euclidian Geometry (Emily)

by Minkyu and Ida

In our class, we focused our attentions on isometries and Euclid's five axioms. In our first class, we discussed the Euclidean axioms in a planar geometry. Then, we related the axioms to the spherical geometry, where lines are great circles on the surface of the sphere. In particular, we showed that while Euclid's first 4 axioms are consistent with spherical geometry, the 5th axiom, which states that lines have parallels which are uniquely defined by a point, turns out to be false on the sphere. We also talked about the isometries on the planar geometry and the spherical geometry. We showed that any isometry can be represented as a composition of rotation and reflection in spherical geometry (and translation in planar geometry). In fact, we showed that translation and rotation are both compositions of reflections, proving that reflections are enough to generate any isometry. Lastly, on the final two days, we talked about the geometry of hyperbolic planes. We defined the hyperbolic plane as the half of a complex plane whose imaginary part is positive. On the hyperbolic plane, lines can either be vertical, or they can be semicircles centered around the origin. We determined the validity of Euclid's axioms in hyperbolic space (again, the 5th axiom turns out to be invalid), and considered the notion of distance in hyperbolic space (vertical distance gets smaller the closer you get to the x axis, while horizontal distance kind of doesn't exist at all).

Matrix Psych (Adam)

by Nicholai

In Matrix Psychology, we explored the inner psyche of matrices. We began by considering the geometric interpretation of the determinant as the signed volume of a parallelepiped described by a matrix A. Then we discussed *stretchy values* λ and *stretchy vectors* $\vec{v}$ which satisfy $\lambda\vec{v} = A\vec{v}$. Every symmetric matrix can be written as $Y^T DY$, where D is its *herm*, a diagonal matrix containing its stretchy values, which stretch the axes. The *eet* Y has columns which are the stretchy vectors, the spinny-reflecty part. Since Y is *orthonormal*, we have $Y^T = Y^{-1}$ which makes calculations very easy. This decomposition is really useful, since we know for example that $A^3 + A = Y^T (D^3 + D)Y$ and $\det(D) = \det(A)$. We concluded by discussing expected values of random matrices of given herms and arbitrary eets, which can be expressed surprisingly nicely.

Don't Have Bad Taste (Connor)

by Adrian

Throughout Don't Have Bad Taste, we dealt with the colorings of the polyps of sea creatures. There exists a very specific set of rules in coloring sea creatures. The first, and by far the most important, rule was that if two polyps share a tentacle, then they must be different colors. Using this rule, you can color any graph, but some are far easier to color than others. We as a class have proved multiple conditions on what numbers of colors are necessary. We tried to find an algorithm for this... and failed miserably. We showed that a sea creature is 2-colorable if and only if it has odd cycles, and we also constructed a pair of bounds for the number of colors necessary for a given sea creature.

We moved on to accounting for non-crossing sea creatures, and discovered not the well-known four-color theorem, but the far easier to prove six-color theorem for planar graphs. Then, being an easily distracted group, we quickly moved on from trying to find out how to color sea creatures to how many ways we can color them. We investigated and found the deletion-contraction recurrence on the number of ways to color, along with a newfound fact that the ways to color a certain graph is a polynomial in the number of colors. Because this recurrence looked very similar to the recurrence for the number of regions defined by hyperplanes from Root, we immediately decided to try to make an explicit connection. We found that plugging in -1 into the chromatic polynomial actually gives the number of regions defined by a set of hyperplanes corresponding to the sea creature.

Algebraic Geometry (without the algebra (and the geometry)) (Brian)

by Pavle

Throughout the week, Brian led us through the basics of algebraic geometry but without most of the algebra and the geometry. We started of by playing with Brian's triangular hats, which had distinguishable labels on each vertex. We found three ways to represent the full space of these hats using three different congruence definitions for triangles: SSS, ASA, and SAS. A notable result was that the space of SSS graphed onto the $3-D$ coordinate plane was an infinite tetrahedron. We also studied the border cases of each of these definitions, finding how each coordinate system defined different degenerate HATs. We then moved onto triangles, finding that its analogous SSS graphed onto $\frac{1}{6}$ of the original tetrahedron. Finally, we talked about acid boxes (Brian's favorite), parallelograms with ordered labels A, C, I, and D. After finding a coordinate system, we realized that they leaked at the corners. So, we set off to find what different shapes of acid box left the same patterns of leaked acid, all the while making sure not to accidentally flip over the box and spill the acid.

Dingoöptimization (Josh)

by Allen

A long time ago in a land far far away (Australia), a young couple emerged triumphant in the lawsuit over the death of their baby. The murderers, the judge correctly ruled, were the dingoes. To promote the common welfare and secure the blessings of liberty to our babies, we created an algorithm to satiate the dingoes' hunger in our mutual relationship with them (we feed them, and in return they don't eat our babies).

Specifically, the dingoes are faced with a plethora of foods F, a set of vectors with each vector having a fixed calorie count. The dingoes want to pick a meal, a subset of F, that is noncrap and maximizes their calorie intake. In turns out that the dingoes can always act greedily: pick the next-largest food that doesn't turn the meal crap. The proof relies on a cool property of food: if there are two noncrap meals, say M_1 of k foods and M_2 of $k-1$ foods, then there is a food f in $M_1 \backslash M_2$ such that $M_2 \cup \{f\}$ is noncrap. Then using this property, we generalized our set of foods by defining an outback as a set F of foods and a set $N \subseteq \{\text{subsets of } F\}$ that satisfy:

- $N \neq \emptyset$
- if $i \in N$, $\forall J \subseteq I$, $J \in N$
- if $T \in N$, $|T| = k-1$, and $T* \in N$ with $|T*| = k$, then there exists $f \in T \backslash T*$ such that $T \cup \{f\} \in N$.

Sample outbacks included: the outback steakhouse (subsets of $\{1, 2, \ldots, n\}$ of certain sizes), the Great-Barrier-Reef (subsets of pairlops of a coral reef), and the vector outback. In fact, many outbacks (but not all) are actually isomorphic to a vector outback! Josh seemed really hype about outbacks and food, almost too excited. The question now is: Is Josh a Dingo?

Magic Functions (Connor)

by Jonathan H

A generating function represents a sequence $a_0, a_1, a_2, \ldots$ by the "polynomial" $\sum_{i\geq 0} a_i x^i$. The Catalan numbers, the Fibonacci sequence, and sequences coming from permutations were studied and many nice closed forms of them were derived. To find these polynomials, students often found a recurrence relation and used it to represent the generating function in terms of itself. Though there were some questionable antics throughout (what's $(n-3/2)!$?) every equation simplified perfectly in the end: truly magical.

Erdős Magic

by Josh and Jessica

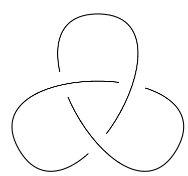
Corrine In our time in Erdős Magic, we looked at existence proofs via Paul Erdős' magical probabilistic method. We found that if the probability of some outcome having a property was less than 1, then at least one outcome in the probability space must not have that property. Using this logic together with identities like the Batman Counting Principle (sometimes known as π) we found magical bounds such as:

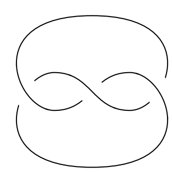
- for nonnegative random variable X and positive a, $\mathbb{P}(X \geq a) \leq \frac{\mathbb{E}[X]}{a}$;
- as long as $\frac{\binom{n}{6}}{2^{14}} < 1$, you can color the tentacles of an $n$tarfish red and blue so there is no one-colored 6tarfish inside of it!;
- bounds for the number of lone bois and big bois in a jellyfish by counting small bois of size 2.

Knot Theory, practice (sarah-marie)

by Dimitri

The very first thing you need to do when working with knots is, of course, know what a knot is. This is where sarah-marie had started in the knot theory class. After coming to the conclusion that a knot is some string where the ends are fused, we drew a series of commonly found knots including the trefoil knot [pictured below, left] and a knot from a randomly tangled from a piece of string [pictured below, right]. A conjecture arose that the two knots shown were the same knot.





To show this, you could draw intermediate diagrams with minimal steps between each diagram. The minimal changes we make to go from one intermediate diagram to another are called the Reidemeister moves. We continued by looking at the (l, m) torus knot which can be created by wrapping a strand around a torus such that is makes l longitudes and m meridians. Eventually, we focused on the problem of proving knots different rather then the same. We did this with the help of an invariant. The FLYPMOTH Polynomial can be found for any knot in a recursive manner depending on orientation of crossings. This polynomial invariant allowed us to show that a trefoil knot is different from its mirror image.

Voting Methods, or why we can't have nice things: a proof (Emily)

by Greg and Frank

This class was all about studying voting systems and their various properties. We started the first day with considering an election for the office of President-Chancellor. Our ballots were ordered lists of preferences, and we found various methods to take a collection of ballots, or a ballunch, and output a result. Given the ballunch Emily provided, we found methods to have each one of the five candidates win, including the method of "emperor", where only one person's vote matters. We then considered the various properties that we'd like in such a method, such as always producing a winner and unanimity. Emily also gave us the lower bound property (if someone wins an election in a given ballunch, just moving them higher will not make them lose) and independence of irrelevant alternatives, or IIA (just moving around person C would not change whether person A is ranked higher or lower than person B) to consider. All of these properties we found were nice, and early in the week we proved a weak theorem on what properties a voting system can have.

We then took a diversion and looked at a method to study how powerful a given person's vote was, looking at how many voting blocs a given voter was important in. The last two days were spent proving Arrow's Theorem, which stated that only "emperor" produces a winner and satisfies the lower bound, IIA, and unanimity. We also showed a similar result for proportionate systems.

In the Long Run (Gideon)

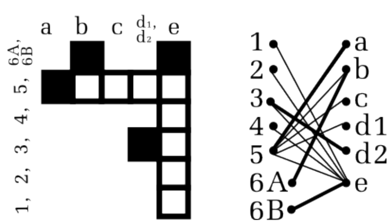
by Cole and Emma

We began by recalling the definitions of events and probability functions. We then applied our knowledge of probability to breakfast food! Given the choice between IHoP and IHoB, we used the probabilities of returning to the same restaurant and of switching restaurants to determine the probability that we'd eat at IHoP on some arbitrary day far in the future. We then extended the IHo restaurant family to include n restaurants, where each day we would move to one of two adjacent IHo's, each with probability 1/2.

We studied random walks on the integers, proving that in any such walk, you will return to where you start with probability 1, which also means that you'll return to where you start infinitely many times. We then struggled to prove the same about taking a random walk on $\mathbb{R}^2$, but Gideon proved for us that, while you will return to where you start infinitely many times on $\mathbb{R}^2$, the same is not true for $\mathbb{R}^d$ for any $d \geq 3$.

Rook Research (Hannah)

by Nikhil



A polyomino is defined to be a subset of an infinite chessboard such that one can use rook moves to move between any two squares of the polyomino. What is the maximum number of non-attacking rooks that can be placed on any given polyomino? We first came up with proofs for the maximum number of rooks on specific polyominoes. In addition, Nicholai found a construction to turn any given polyomino into a two-columned sea creature in which each vertex is a column or row and each edge corresponds to a box. Using this construction, we proved Henry's claim that given r rows, if a subset of n rows intersect with m columns, and $m < n$, then at most $m + r - n$ non-attacking rooks may be placed (assuming there are no more rows than columns). Hannah then showed us the algorithm of "contagious wiggling." First, we find left vertices that have no rook edges. Wiggling moves left to right along non-rook edges, and right to left along rook edges. We keep wiggling until we reach a right vertex, called p, that doesn't have any rook edges. Tracing back from p along our 'wiggling path' creates a path from p to one of our original vertices. By turning the rook/non-rook edges to non-rook/rook edges on our path, we increase the number of rook edges by one. Adrian then showed that this algorithm only stops when we have the maximum number of rooks, making use of Henry's earlier condition. We showed that if a sea creature isn't connected, then the polyomino that corresponds to it can't be connected either. However, Jessica and Albert showed that the converse is not true.

Math Saves The World: Combinatorial Optimization (sarah-marie)

by Ian C. and Angela

How do you fix the problems in this world? Easy: with math! Looking at examples such as assigning note-takers in class, using positive bag match at airports, and bringing essentials with you on your trip to the moon, we showed how math enables us to optimize such tasks. Optimization requires several elements: first, we need to figure out what specifically the problem is (identification). After that, we have to obtain the relevant data (collection). Finally, we have to combine these into a mathematical model. This model should capture the variables involved and all of the constraints. One specific problem we focused on was about note taking assignments, and we searched for a model to decide who should take notes when while keeping in mind constraints such as that nobody can take notes twice. We also discussed what we would pack to a spaceship, which is a specific example of the Knapsack Problem. Here, we had to keep in mind the maximum weight or volume the spaceship can hold. Finally, the model needs an objective function, which takes as input what we arrange or bring (subject to the constraints) and returns a value depending on how good the proposal is. This objective function, when maximized, tells us the best solution.

Projective Geometry in Perspective (Tom)

by Ian S

Every student at MathILy got a taste of projective geometry from Rebecca RG's Daily gather during week one. In Tom's Week of Chaos class, we explored what makes the projective geometry so uniquely useful and interesting. We first discussed how projective geometry is one of the only areas of math that was motivated by art. Realistically portraying a 3-dimensional image on a 2-dimensional canvas requires an understanding of how certain points project to other points. In projective space, "points" of the form (x, y, z) actually describe the line composed of all points (kx, ky, kz). For that reason, n-dimensional projective space, denoted $\mathbb{RP}^n$, can be thought of as the set of all lines through the origin in $\mathbb{R}^{n+1}$. Similarly, "lines" in projective space can be described by equations of planes through the origin in the form $ax + by + cz = 0$, abbreviated $[a, b, c]$. The algebra of points and lines works completely identically, and the axioms of projective geometry do not differentiate between them, so any theorem about points, lines, and their intersections holds true when points and lines are swapped. This concept, called duality, comes in handy for many geometric proofs.

We also considered projections between two planes inside $\mathbb{R}^2$. When we project from a subject plane to a canvas plane, we simply map every point (x, y, z) in the subject plane to its corresponding point (kx, ky, kz), where k is chosen to make the new point lie in the canvas plane. The only problem is that some points in the subject plane don't get mapped anywhere, because their lines through the origin don't intersect the canvas plane. We call these points the points at ∞ relative to our subject plane. Similarly, some points in the canvas plane will have no points mapped to them, which we'll call the points at ∞ relative to the canvas plane. Mind-blowingly, we saw that the space of "points at infinity" in $\mathbb{RP}^n$ is equivalent to $\mathbb{RP}^{n-1}$, meaning $\mathbb{RP}^n$ can be written $\mathbb{R}^n \cup \mathbb{RP}^{n-1}$.

Characters of Sprougs (Josh)

by Margaret

After a brief review of sprougs and their isomorphisms, we delved into *morphisms* (mappings which preserve sproug operations) and in particular *representations*, or morphisms from a sproug to the sproug of $n \times n$ invertible matrices of complex numbers. We considered various representations of the sproug of symmetries of an equilateral triangle and investigated when representations were isomorphic. We discovered that the *Frank sum* of representations (a diagonal concatenation of matrices) is itself a representation. We also found that a representation is isomorphic to a nontrivial Frank sum if and only if it has nontrivial invariant subspaces which span $\mathbb{C}^n$ and intersect in 0. We then studied irreducible representations, which have no proper nontrivial invariant subspaces. We contemplated *stretchy vectors* and of course *characters* (functions $\chi_\rho : S \to \mathbb{C}$ that map each $s \in S$ to the *trace* of $\rho(s)$), leading to the startling conclusion (which we unfortunately did not have time to prove completely) that a finite sproug S has finitely many irreducible representations.

Surreal Numbers (Corrine)

by Pranav

Alice and Bill wanted to escape with a romantic getaway, but instead they ended up finding an ancient Hebrew tablet. On this Hebrew tablet, Conway defined numbers. From this, the class determined all integers and some other properties of Conway's numbers, and they were good. We then found out that Alice and Bill discovered another piece of the tablet. On this Conway defined addition, subtraction. From this, the class proved further statements about the equality of numbers and the commutative and associative properties, and they were very good. The class then defined multiplication and the numbers were fruitful and multiplied.

We defined very big numbers like Alepha (denoted by a whale) and very small numbers like Zhao and Wang. And the rules were good. They were very good.

Origametry (Tom)

by Katie

The first thing we learned in this class was how to make boxes to put origami stuff in. Then we made PHiZZ units, and learned how to make buckyballs out of them. We calculated how many units we would need - the smallest number is 30, but they can be WAY bigger. (Tom made one out of 810 units!) We learned a strategy for colouring them with 3 colours so that no 2 of the same colour touch each other, by making a super-supersoaker. Then we folded polygons and parabolas and learned to trisect the angle. We found a way to solve any cubic equation by folding lines. At the end of the last class we got to look at some really mind-blowing things that Tom made.

Algebra, A Singular Passion (Josh)

by Ian C.

The main topic of this class was the concept of *georges*. We call a set of elements with a binary operation a *george* if the set is closed under the operation, the set contains some identity I for the operation, and if each element in the set a has an inverse a^{-1} such that $aa^{-1}a^{-1}a = I$.. Examples of georges include the symmetries of a triangle under composition and the rationals under addition. Interestingly, sets that are george can have subsets that are *also* georges in and of themselves–we call such sets *sub-georges*. The rationals of the form $\{\frac{a}{2^n} \mid a, n \in \mathbb{Z}\}$ is such an example. We then looked at qualities of georges–for example, a subgeorge $H \in G$ is *henry* if for all $h \in H$ and $a \in G$, the element aha^{-1} is in H. As it turns out, such definitions can even apply to sets of sets. That is, we can take some george G, take subsets $A_1, A_2...A_n$ of G, and consider whether $A_1, A_2...A_n$ is itself a george under the right operation. Things get meta quickly.

Computers: What can they do? Can they do things? Let's find out! (Gideon)

by Danielle

On Monday, we found an algorithm that could find the minimum cost system of roads between embassies that had at most $m^2 + m(n-1)$ steps, where n is the number of embassies and m is the number of roads. Then, we learned about Big O (Asymptotic Notation.) Afterwards, we considered the following questions: Given a boolean formula, can a computer find a solution? Can a computer recognize no solutions? A boolean formula is in *k-Andor form* if clauses contain k terms. Given a k-Andor formula, does a satisfying assignment exist (k-Sandor)? When is k-Sandor hard? When is k-Sandor harder than ℓ-Sandor? Adam posed the question: What's harder: being a millionaire or buying a Ferarri? Buying a Ferrari is easier because becoming a millionaire allows one to do so. Similarly, with a solution to n-Sandor, we can solve k-Sandor if $n \geq k$.

We then considered problems on graphs, including finding a tarfish (given a graph, find k joints connected by bristles), a quarfish (given a graph, find k joints such that none are connected by bristles), or a shadowfish (given a graph, find k joints such that every bristle touches one of these joints). We proved that the tarfish and quarfish problems have the same level of difficulty.

Cantor Set (Hannah)

by Eric

Begin with the interval $[0, 1]$. Take out the middle third, leaving you with $[0, \frac{1}{2}] \cup [\frac{2}{3}, 1]$. Take out the middle third of each of those segments. Define a pseudo-Cantor set PC_n as the result of n iterations of this removal, and the Cantor set C as the intersection of all pseudo-Cantor sets. This brought some conjectures. Are the only points in the Cantor set endpoints of middle third segments? Is the Cantor set uncountable? What is the length of the Cantor set? Through some close examination of examples, we proved that a number is in the Cantor set iff its ternary (base 3) expansion contains only 0s and 1s. This helped us show that the Cantor set has length 0, is uncountable, and contains more than just endpoints. Next up we generalized the Cantor set by changing the recursive step. We can:

- Take out a different fraction x from the middle, obtaining xC.
- Take out $\frac{1}{3}$, then $\frac{1}{9}$, etc.
- Split each segment into $2n+1$ segments and take out the even numbered segments.

We tried to prove limit-preserving bijections between them, or bijections such that every sequence in one set which converges to a limit also converges in its image in the other set. By using a slick argument where we assign addresses to each point based on whether they are in the right or left segment at each iteration, we managed to prove that there is a limit-preserving bijection between C and xC.

Integrate Like the French (Brian)

by Allen

Suppose you had a collection of pennies, nickels, and dimes in a line. You challenge your French friend to count the change as fast as possible. While you go down the line, adding the next coin's value to your total count, the Frenchie scans the coins, counts the number of pennies, nickles, and dimes (p, n, and d), and tells you that the total change is $p + 5n + 10d$ cents. Of course you both get the right answer, but whose way was better? After rigorously defining upper and lower Darboux sums and reviewing Riemann integration, we developed Lebesgue integration, the analogue of the Frenchie's coin-counting scheme for functions. To do so, we introduced the idea of open sets, open measures, and measurably. Finally, similar to Riemann integration, we defined the Lebesgue integral using upper and lower integrals of functions that are easily Lebesgue integratable: characteristic functions of Naughty boys. And thus, one of the functions not integratable in the Riemann world becomes one of the clearest ones for Lebesgue: if $f : [0,1] \to \mathbb{R}$ satisfies $f(x) = 1$ if x is rational and $f(x) = 0$ otherwise, then

$$\int_0^1 f \, \mathrm{d}m = 0.$$

Loops on Loops on Loops (Gideon)

by Corwin

This class was based on the general scenario of a person who lives at iHoP and travels on a loop, leading back to iHoP. Two loops are "the same" if it is possible to drag one into the other while staying in the space. Sets of loops that are the same are called loop types. The class considered **C**rouges o**U**t of **L**oop **T**ypes (CULTs), where "crouge" is a portmanteau of the words "crow," "sproug," and "george." We looked at CULTs in different timelines. For the first timeline, the town started as an infinite plane with iHoB removed, meaning that loop types are based on the number of times that a loop circles iHoB before returning to iHoP. However, in the next couple of years iHoB began to take up a large area, and the town built a wall on the border to limit this expansion. By 2019, only a circle remained, but the CULTs of 2017, 2018 and 2019 were all isomorphic. However, in the year 3030, iHoB was spherical, and it was possible to walk anywhere on the sphere, meaning that there was only 1 loop type. The other timeline was where the town was an infinite helix in 2017, but collapsed into a circle in 2018. There was a one-to-one correspondence between the paths 2017 and 2018. However, since a cycle in 2017 results in traveling up or down, loops in 2018 might just be paths in 2017. For this reason, students conjectured that loop types in 2018 could be distinguished by the ending point of the corresponding path in 2017.

The Ocean Is Big And Blue (Corrine)

by Albert

The class first started by examing Reef (Ramsey) numbers $R(n, k)$. Given a reef where the tentacles are 2-colored, $R(n, k)$ is the number of polyps required to guarantee an n-tarfish of the first color or a k-tarfish of the second color. We proved several properties to obtain a possible upper bound for $R(n, k)$. We then defined more general Ramsey numbers $R_t(n, k)$, where we make t the number of polyps that a tentacle can contain, and also $R_t(k_1, k_2, \ldots, k_n)$ where now we have n colors. We proved a few more properties, most notably the Erdős – Szekeres (Happy Ending) Theorem, which we proved that if there is $R_4(5, n)$ points in the plane, then n of them form the vertices of a convex n-gon. We also proved Schur's Theorem, which states for any r there exist a value n such that if $\{1, 2, 3, \ldots, n\}$ is colored with r colors, then there exist x, y, z in the set $\{1, 2, 3, \ldots, n\}$ such that x, y, z, are the same color and $x + y = z$. While proving these propertie, we repeatedly simplified harder problems into problems that we have solved before, which means that we related the new problems to the initial problem of proving that $R(3, 3) = 6$. This is a technique that can simplify problems to make it easier to solve, which was the theme of the class. Different ways to simplify problems is very common in the class, including when we proved theorems on Party (van der Waerden) numbers.

p-adicts Anonymous (Brian)

by Henry

The p-adics rely on a new definition of distance, so to get comfortable with what distance means in general, we first looked at a way to construct the reals from the rationals with our normal definition of distance. Each real number can be thought of as a series a_n of rational numbers that converge to it, with the series adhering to the following "closeness condition:" For any arbitrarily small rational number $r > 0$, there must exist some term a_k after which all the terms of the sequence are within r of one another. This closeness condition relies heavily on our standard definition of distance between two real numbers, the absolute value of their difference. To construct the p-adics, we defined a new notion of distance as follows. For some integer a, $||a||_p$ (pronounced "p-norm of a") is the reciprocal of the largest power of p that divides a. For example, $||4||_5 = 1$, $||5||_5 = \frac{1}{5}$, and $||18||_3 = \frac{1}{9}$. Two nice properties hold: The p-norm preserves multiplication, i.e. $||a||_p \cdot ||b||_p = ||ab||_p$; and $||0||_p = 0$ for all p. The p-norm can be extended beyond the integers to the rationals by defining $||\frac{a}{b}||_p = \frac{||a||_p}{||b||_p}$. For example, $||\frac{4}{5}||_2 = \frac{1}{4}$.

With our new notion of distance, we defined the p-adic numbers in a manner similar to the reals: as series subject to a certain closeness condition. To represent a number n in the p-adics, we generate a sequence of digits in base p $(a_k a_{k-1} ... a_1 a_0)_p$ such that for every rational number r, there exists a k such that $||(a_k a_{k-1} ... a_1 a_0)_p - n||_p < r$; that is, the "distance" between the sequence and n can be made arbitrarily small. We developed an algorithm to generate p-adic representations of rational numbers, and found that all such representations eventually repeat their digits. Many numbers can be represented this way, including some square roots! We generalized our algorithm for the rationals to handle p-adic representations of square roots, then generalized even further to find general roots of polynomials. It turns out it's much easier to find roots of polynomials in the p-adics than in the reals, despite the difficulty of everything else about the p-adics.

Fancy Note-Taking Diagrams (Hannah)

by Jonathan H

In this class, we worked with a different type of note-taking diagram, where there are only horizontal lines between the vertical stalks that sort the two names currently in those stalks alphabetically. Students first considered algorithms that sort a collection of boxes by repeatedly weighing two of the boxes at once, and used Big-O notation to determine approximately how many weighings would be needed. The decided algorithm was a merge sort, where by using the knowledge that two collection of boxes were already sorted, there is a nice way to efficiently sort the two collections together. Then they moved on to creating a network of horizontal lines on a note-taking diagram that would sort a list of names in any order, eventually settling on a nice even-odd merge sorting method called Batcher merge.

iCalculus>Calculus (Connor)

by Zach

In iCalculus>Calculus, we extended concepts from single-variable calculus on the reals to calculus in the complex plane. We started by defining complex functions in terms of functions from the reals and coming up with notions of limits and differentiability. We discovered that, informally, a function in the complex plane is differentiable if and only if the derivative in the real direction and the derivative in the imaginary definition match. Then, we explored integrability, defining integrating along a curve in terms of real functions. We discovered that if a complex function was differentiable on and in the interior of a loop, the integral along the curve was 0. Exploring integrals further, we discovered several beautiful results about complex functions, including an explicit formula for the value of a differentiable function inside a circle in terms of the values it attains on the circle. We found that we could determine the Taylor series for differentiable complex functions and that every differentiable complex function was differentiable infinitely many times with a power series that converges to the value of the function. This contrasts to the real world, where some functions behave very poorly (if we define $f(x) = e^{-1/x^2}$ for nonzero x and $f(0) = 0$, the power series of f is 0 at the origin, even though the function attains 0 only at 0).

Mind Reading (Emily)

by Charles

Mind reading is only possible using quantum computers, not classical computers. This is because mind reading is clearly equivalent to breaking RSA encryption, which reduces down to finding the prime factors of an extremely large integer. A feasible algorithm with polynomial runtime, published by Peter Shor in 1994, is only possible with quantum computing. We began by reading a basic blog post which explained the general process of the algorithm with minimal terminology. Then, we studied Shor's original paper to understand the mathematical manipulations that make his algorithm work. We discussed the modular arithmetic used to facilitate prime factorization, the implementation and purpose of the quantum Fourier transform, and how to ultimately extract the prime factors after Shor's algorithm is performed. Along the way, we also learned the difference between a qubit and a bit, about quantum superpositions and gates, and the significance of polynomial versus exponential runtime.