Contents

Aaron's Projects

Interpreting Heunen's Axioms for Quantum Mechanics. (Aaron)

Description: In 2009, Chris Heunen gave a list of conditions on a physical theory which guarantee that it's a subtheory of quantum mechanics. He didn't phrase his results in those words, though: what he really gave was a list of conditions which guarantee that a category will embed in the category of Hilbert spaces and bounded linear maps.

The goal of this project is to look for physical interpretations of Heunen's conditions, in the hope of turning his result into an interesting statement about the foundations of quantum mechanics.

Structure: Individual or small group. Research with regular meetings.

Expected Input: Up to you.

Expected Output: If we're lucky, maybe a little progress on an extremely hard problem with no clear win condition.

Difficulty: 🌶🌶🌶🌶

Prerequisites: Category theory, quantum mechanics.

What's Up with Cutting Sequences? (Aaron)

Description: Cutting sequences—the sequences of horizontal and vertical "cuts" generated by a ray of irrational slope drawn over an integer grid—are an easily accessible entryway to a very deep vein of ideas. If you'd like to learn more about them, this project is for you!

I'm no expert on cutting sequences, so we'd mostly be learning stuff together. Depending on your preferences, we can either stick to the books or think about questions that I haven't found straightforward answers to in the literature.

Structure: One-on-one or small group. Depending on your background, we'd probably start out with me teaching you standard stuff about cutting sequences, and then move into a more research-like mode as we reach the end of my (very limited) understanding.

Expected Input: Up to you.

Expected Output: More knowledge about cutting sequences than we had when we started, and maybe even answers to some questions that we can't find answers to in the literature.

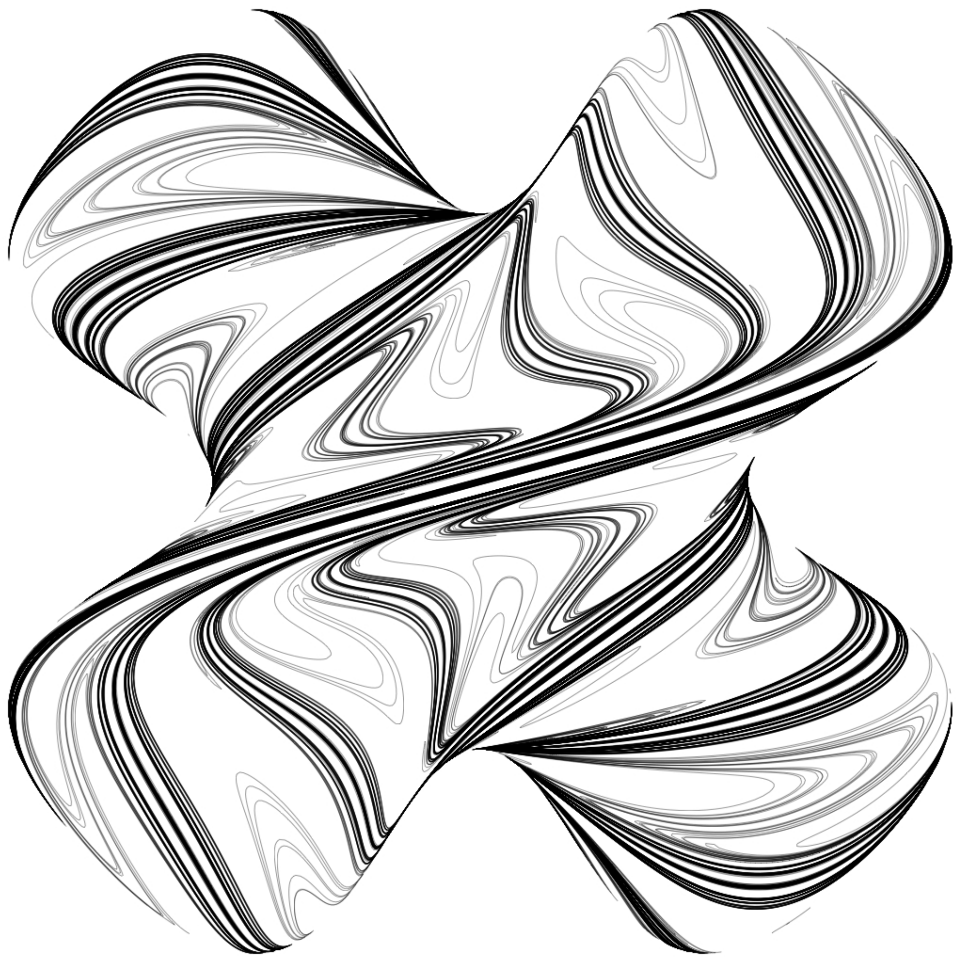
Difficulty: 🌶🌶

Prerequisites: None

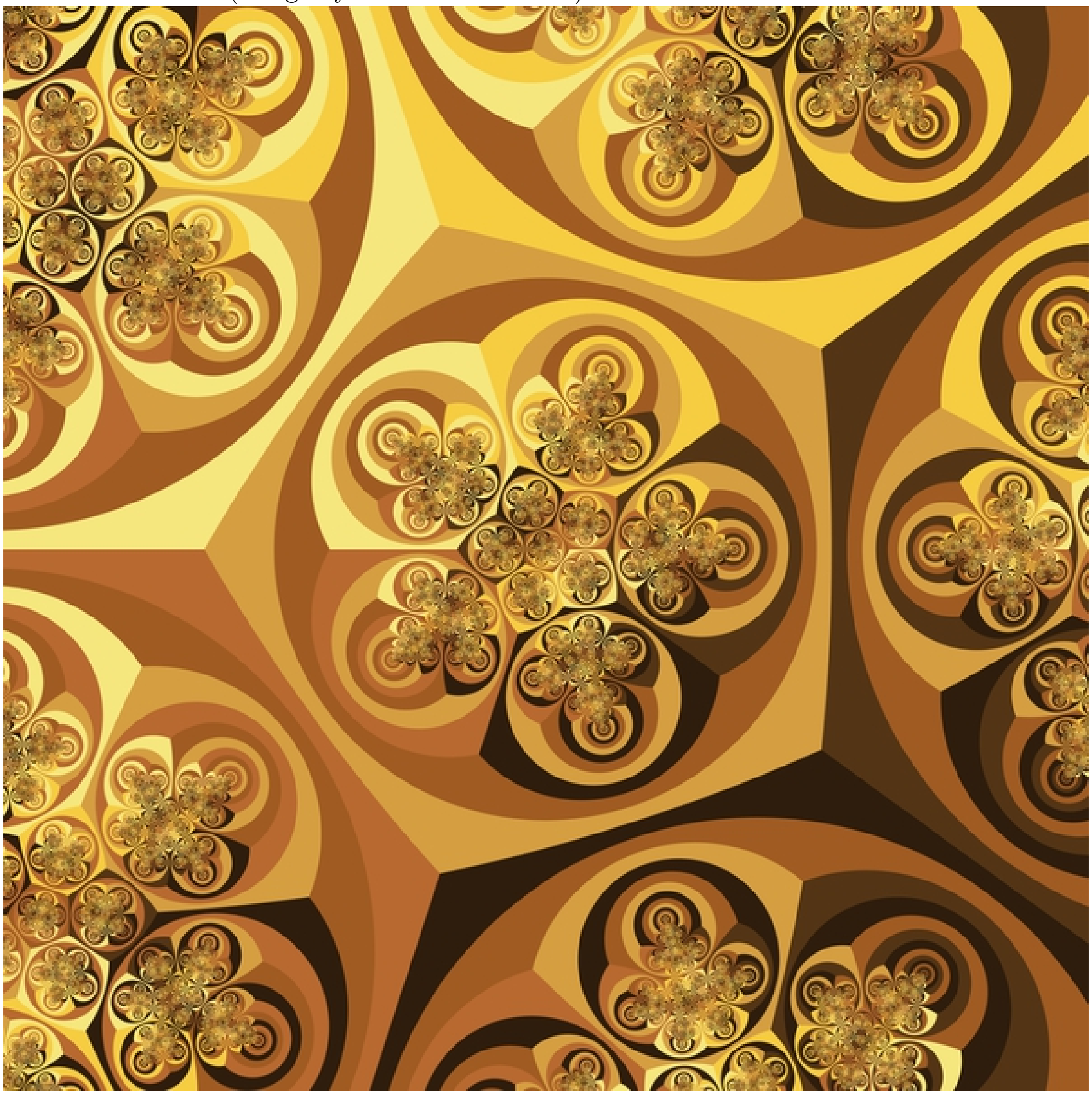
Aaron and Kevin's Projects

Let's Make Rad Art! (Aaron, Kevin)

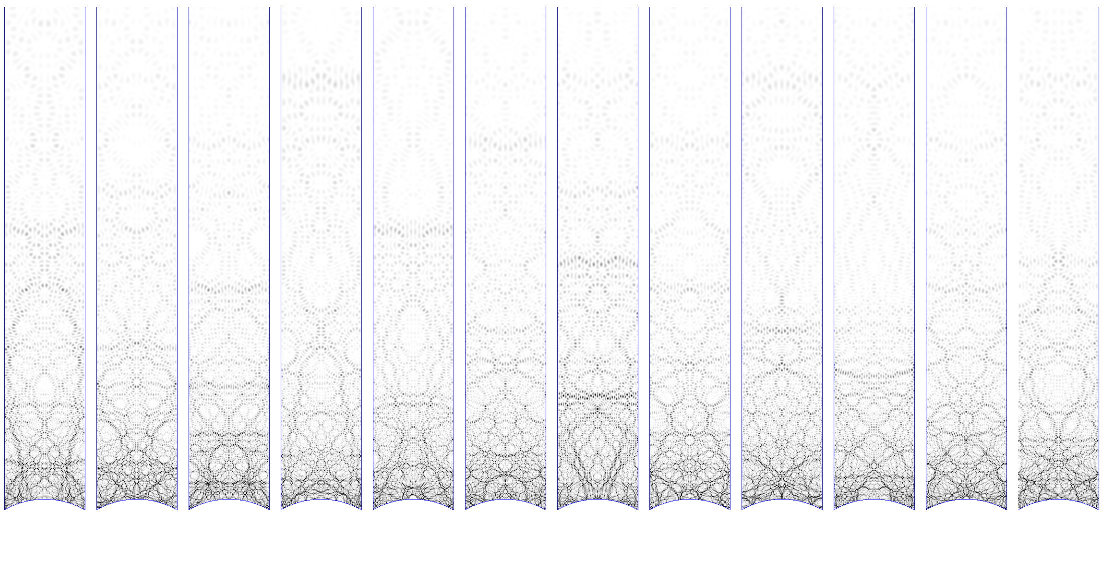
Description: Do you want to make something like this? (Image by Curtis McMullen.)



What about this? (Image by Vladimir Bulatov.)



I don't even know what this is, but I want it! (Image by Alex Barnett.)



This project is a space for you to make rad math art with computers, individually or in groups.

Structure: You make art! Kevin and Aaron provide inspiration, math help, and programming advice.

Expected Input: Source code.

Expected Output: Rad art to show off at the project fair.

Difficulty: 🌶🌶

Prerequisites: None

Alfonso's Projects

Π_1(**Dorms**). (Alfonso)

Description: I stand in the main lounge holding one end of a very long rope. You grab the other end, you go running through all rooms of Platt and Howard, and then you come back. I then pull both ends of the rope as tight as I can. Can you describe all the possible loops that the rope may end up in?

This question is exactly the same as calculating the fundamental group of the dorms. (If you do not know what this means, you can learn it in Jeff's Week 3 class).

Structure: You have to be able to work without much supervision. I will help you get started. After that, how much or how little you do depends only on you. We will meet when you have something to tell me or when you decide you want some help or when you simply want to discuss your ideas, always at your initiative. If you are not a good independent worker, the project will go nowhere.

Expected Input: 4 to 12 hours total

Expected Output: A poster presentation at the project fair

Difficulty: 🌶🌶

Prerequisites: It will help to know what a group is and to know what the fundamental group is.

Projects in Combinatorial Game Theory. (Alfonso)

Description: Here are various combinatorial games you could analyze:

- **Game of Euclid.** (🌶)
  Two players start with a pair of distinct positive integers, a and b. The first player subtracts any positive multiple of the smaller integer from the larger one, leaving a positive remainder. She thus creates a new pair of positive integers. For example, if the initial pair were $(100, 13)$, the first player could choose to replace 100 with $100 - 4 \times 13 = 48$, leaving the pair $(48, 13)$. The second player repeats this process with the new pair of integers. The players continue taking turns in this fashion. The player who is able to create a pair of equal integers wins. Who has a winning strategy?
- **Mira's game.** (🌶)
  We start with two bags of M&Ms, one with 19 and one with 20. On her turn, a player has to eat all the M&Ms in one of the bags, and then split the M&Ms from the other bag between the two bags, leaving at least one in each bag, not necessarily evenly. The she gives the two bags to the other player. For instance, the first player may eat the 19 M&Ms and divide the other 20 by putting 12 in one bag and 8 in the other. The player who receives two bags with one M&M each loses as she can no longer move. Find the nim value of every position.
- **The Princess and the Roses.** (🌶🌶–🌶🌶🌶)
  Princess Alice has two suitors: Laura and Renata. They alternate days in trying to gain her heart by bringing her roses. They pick up the roses from the same garden, which has roses of k different colours. Each day, the corresponding suitor will bring her one or two roses, but never two roses of the same colour. The suitor to pick the last rose from the garden will win Alice's heart. Who will succeed?
- **Variants of Nim.** (🌶–🌶🌶🌶🌶)
  You already know how to play nim. Here are three variants, from easy to very hard:
  - Ordered Nim. We play with various piles of berries that need to have different sizes. When we remove berries from one pile, we may not allow that pile to become smaller than or equal to any pile that initially had less berries than it.
  - Antonim. We play with various piles of berries that need to have different sizes. At no time are two piles allowed to have the same number of berries, except that we are allowed to have multiple piles with 0 berries.
  - Simonim. We play with various piles of berries like in nim. However, if there are multiple piles that have the same number of berries, in your move you may choose to take berries from various of them, as long as you take the same number of berries from all the ones that you take berries from.
- **Histeresis Nim.** (🌶🌶)
  This game is similar to nim. We play with a single pile. On his turn, the first player may remove any number of beans, at least one, but not the whole pile. After that, each player has to remove a positive number of beans no greater than twice the number of beans removed by the previous player. How do you win?
- **Heads Turn.** (🌶🌶🌶)
  We start with a row of coins, some are heads-up and some are tails-up. In your turn, you remove one of the heads-up cards, and you flip its immediate neighbors, if any. We take turns doing this. The first player who cannot move, loses. Finding a complete winning strategy is an open problem, but there are lots of partial results awaiting you.

In this project you will work on trying to solve one of the above games. A complete solution with proof is desirable. Some cases are too hard (or even open) but there are plenty of partial results (for example, solving the game in a particular case) that can be very interesting by themselves.

Structure: Please indicate in the form whether you want to work alone or in a group, and whether you want to use coding. You have to be able to work without much supervision. I will help you get started. After that, how much or little you do depends only on you. We will meet when you have something to tell me or when you decide you want some help or when you simply want to discuss your ideas, always at your initiative. If you are not a good independent worker, the project will go nowhere.

Expected Input: The (🌶) game can be solved in 4 to 8 hours total. For the others you should expect to put 8 to 20 hours total.

Expected Output: A complete solution for the game, a poster at the project fair (together with a stand to demonstrate the game and play with other campers).

Difficulty: 🌶–🌶🌶🌶🌶

Prerequisites: It will help if you took Mira's Combinatorial Game Theory class this year (or Alfonso's class in a past year). Some of the games will be helped by coding.

Square Dance Calling. (Alfonso)

Description: If you attended my colloquium this week, you learned a bit about square dancing. Performing the dance is very mathematical, but calling it (and writing choreography) is even more so. Your choreography needs to be danceable, resolve, flow smoothly, and have good timing.

Good square dance callers can *sight-call*. This means they can improvise and make up, on the spot, a bunch of calls that work well as a choreography and shuffle the dancers, and then, on demand, they can unshuffle the dancers from any configuration and resolve. The process is not unlike cubing.

Are you ready for the challenge?

Structure: I will select a collection of calls that give you flexibility and work well together for you to learn. I will also give you a link to the whole list of calls (in case you need more), to online animations, and to some software that allows you to try and play choreographies. Your job will be to learn as much as you can and to try to write your own choreography and then call it. If we get enough takers, you can call for your friends. If you are really daring, we can play a game where I scramble the dancers and you unscramble them. We will meet to discuss your progress and practice regularly.

Expected Input: I suspect it will take a fair number of hours to get to a decent outcome, maybe about 24.

Expected Output: A demonstration at the Talent Show or at the Project Fair

Difficulty: 🌶🌶🌶

Prerequisites: None

The Coin Game. (Alfonso)

Description: Start with various piles of coins. Then:

(1) Order the piles from largest to smallest.
(2) Take one coin from each pile and put them together into a new pile
(3) Go back to Step 1.

What happens? There is a plethora of interesting questions to ask about this process, and many little combinatorial results to discover (and prove!)

Structure: You have to be able to work without much supervision. I will help you get started. After that, how much or little you do depends only on you. We will meet when you have something to tell me or when you decide you want some help or when you simply want to discuss your ideas, always at your initiative. If you are not a good independent worker, the project will go nowhere.

Expected Input: It will take you 4 to 8 hours to come up with many interesting results.

Expected Output: A poster at the project fair

Difficulty: 

Prerequisites: None

The Colour of Money. (Alfonso)

Description: The Colour of Money was a short-lived TV show in the UK (you can find it on youtube) that is much more interesting to analyze mathematically than most other TV games. Due to its rules, it necessarily has one unique optimal strategy, although no contestant ever came close to it (or even seemed interested in trying).

Here are the rules of the game (or rather, a game equivalent to it). We have a deck of cards numbered 1 through 20. Before the game starts you receive a target (which will be a number between 50 and 80). Now you draw a card face down, choose an integer number N, and flip the card over. If the number on the card is $\geq N$, you win N points; otherwise, you win 0 points. You repeat this process for a total of 10 times. If you manage to accumulate at least as many points as your target, you win. Otherwise, you lose. What is your strategy?

You can brute-force the problem with a computer, but that will only be the first step. Does this give you a strategy that is easy to memorize, explain, and implement in real time? Let's say that you are at the actual TV game without your computer. What will you do? The goal is to come up with a strategy that is "good enough" and that can be realistically be played by a human.

If you do manage to find a strategy that is easy to describe and implement, can you prove analytically that it is optimal (or at least good enough)?

A second question is to try to calculate what is the probability of winning (with the optimal strategy) as a function of the target.

Structure: You can play individually or in a small group. If there are multiple ones of you interested in this, I may make your strategies compete against each other. As for how we will spend our time, I will help you get started. After that, how much or little you do depends only on you. We will meet when you have something to tell me or when you decide you want some help or when you simply want to discuss your ideas, always at your initiative. If you are not a good independent worker, the project will go nowhere.

Expected Input: See structure

Expected Output: A poster and a demonstration at the project fair.

Difficulty: 

Prerequisites: Coding. This project will require a combination of brute force (for data generation) and clever thinking (for conjecture making).

Alfonso and Jeff's Projects

Contra. . . and Yarn! (Alfonso and Jeff)

Description: Suppose you are a topologist, and you are trying to learn contra dancing. Because you have a hard time remembering all of the moves, you decide to attach some yarn to each of the dancers. As they walk around, the yarn unspools, recording their dance moves on the floor.

After a night of reckless dancing, you walk over to the piles of yarn lying on the ground. Unfortunately, you haven't come up with a system to recover the dance movements from the piles and piles of yarn on the ground. Is it possible that two different dances have the same yarn configuration?

Then you remember, you are a topologist! What you are really interested in is Contra Dance up to Yarn Equivalence! Still, there is a lot for you to think about. When do two contra dances give the same piles of yarn, and what systems can we come up with for classifying these dances?

Structure: 4 campers, working several hours a week to learn about contra dances and classify them topologically.

Expected Input: You will spend 3–4 hours a week, coming up with a mathematical description of when two contra dances are the same and choreographing.

Expected Output: Classify all contra dances up to yarn equivalence. For fun, come up with a dance which is fun to dance, but cannot be distinguished from the trivial dance, and present this through performance.

Difficulty:

Prerequisites: None

Andrew's Projects

Reading Project in Quantum Mechanics. (Andrew)

Description:

> *"I think I can safely say that nobody understands quantum mechanics."*
>
> —Richard Feynman

While it may seem like classical mechanics governs our everyday experiences, it's actually quantum mechanics that turns the hidden gears of reality. This perplexing theory shapes our understanding of the physical underworld; without it, we could not understand the forces that hold atoms and molecules together. However, quantum mechanics also leads to results that are very far from ordinary experience. When you're not interacting with an electron, its motion is only determined probabilistically. The Heisenberg uncertainty principle implies that its position and velocity cannot be simultaneously and precisely determined. And, according to the principle of superposition, an electron has both spin up and spin down until we observe it; then, its wavefunction collapses into one of those two eigenstates.

Structure: 2 meetings/week for about 1 hour each, 5 hours/week of reading on their own.

Expected Input: We'll be reading from a number of texts on quantum mechanics and making a poster/presentation as we go. The exact topics will depend on student interest, but may include the Schrodinger Equation, Hilbert spaces, spin of electrons, how the Heisenberg uncertainty principle follows from the fact that certain operators don't commute, Bell's inequality ruling out hidden variables, entanglement, and quantum computation. I expect to meet with you twice a week, and you need to spend about an hour a day reading and talking with your other group members.

Expected Output: A poster presentation at the Project Fair

Difficulty:

Prerequisites: Single-variable calculus, linear algebra, and a good high school physics class on classical mechanics

Don's Projects

Insider Trading for Beginners. (Don)

Description: Let's say you've got a friend who is clairvoyant: they know exactly which stocks are going to go up and down, and they know it ahead of time. In this case, your investment strategy is fairly straightforward: figure out the biggest percentage increase each day, and throw all your money onto that stock.

That's fairly boring, and also extremely unlikely. What might actually happen, though, is that you have a friend with an ear to the ground at lots of different publicly traded companies. You can't be sure how any particular investment will turn out, but you can assign probabilities to outcomes, and these probabilities sometimes differ from how the other participants in the market place them (in part because those other participants aren't interested in breaking the law).

Given this level of foresight, what is the optimal investing strategy? If a stock has a 90% chance of doubling and a 10% chance of becoming worthless, you should clearly buy some, but how much? If you gamble your entire bank account, you face a 10% chance of going bankrupt, but if you risk too little, you've missed out on a major opportunity.

It turns out that there is a "correct" answer, albeit not one without some amount of controversy. Some prestigious investment firms are adherents of this kind of thinking, although they base their percentages on models that they believe to be better than everyone elses', rather than on illegal inside information.

Your mission, should you choose to accept it, is to find this "best" investment strategy. Then, time permitting, you'll explore variants of this problem in which different investments require different amounts of time in order to come to fruition.

Structure: 1–4 campers, working either together or in parallel on related parts of the problem. I know the answer to the basic version, so I can provide guidance there, but the generalizations to time delays will be as new to me as they are to you — I know some potentially fruitful avenues, but you'll be the one going down them.

Expected Input: Some time spent every day, working on successively more challenging versions of this problem.

Expected Output: A collection of strategies and formulae and a poster at the Project Fair.

Difficulty:

Prerequisites: Calculus, Partial Derivatives

Universal Algebra via Category Theory. (Don)

Description: Did you take Universal Algebra in week 1 and walk away thinking, "this just wasn't general enough"? Are you currently taking Category Theory, and despite how awesome it is, you wish it were more algebraic? Are you devoted to functional programming, so much that you want to learn the deep categorical ideas that make it tick?

If you answered yes to any of these questions, then I've got the project for you! A monad is a special kind of functor T from a category X to itself, together with natural transformations $\eta : 1_X \to T$ and $\mu : T^2 \to T$ satisfying certain axioms. Any pair of adjoint functors gives a (not necessarily unique) monad, and by taking the terminal object in the category of adjunctions giving a particular monad, we can associate a monad to any universal algebra. This construction is more powerful than those used in regular universal algebra, since it applies not just to categories we normally think of as "algebraic," but even to some topological categories, like the category of compact Hausdorff spaces.

Structure: 1–3 campers, reading and working on exercises.

Expected Input: 6+ hours per week; reaching Beck's Theorem before the end of camp will be a significant task.

Expected Output: A thorough understanding of Beck's Theorem, which proves the above results.

Difficulty:

Prerequisites: Category Theory (either my class, or having seen adjunctions before).

Don and Kevin's Projects

Smash Analytics. (Don and Kevin)

Description: The world of sports analytics has recently been experiencing a boom — popular articles appear every day extolling the virtues of using math and statistics in order to understand sporting events, and some of the most cutting-edge mathematical models in the world are those being used to understand Football, Basketball, Soccer, and Baseball.

One important sport has fallen tragically behind: the honorable art of Super Smash Bros. An activity that is filled with data — number of lives, damage dealt and received, time of survival — has, to this point, only been analyzed using after-the-fact data of wins and losses, and then only to create a descriptive ranking of how different characters have performed in tournaments.

This is where you come in: by digging deeper into the data of Smash Bros gameplay, you will create a prescriptive model of performance, whose goal is to not just describe who has won previous matches, but rather to do better than past winning percentage at predicting the outcome of future matches.

Structure: Ideally, the group size will be between 2 and 6. First, we'll build a simple model, around a fake data set. Then, we'll do our best to obtain as large a data set as possible and try to build a sophisticated model around it.

Expected Input: In order to create a nontrivial model, you should expect to spend around an hour per day, most days of the week.

Expected Output: A predictive model.

Difficulty: 

Prerequisites: Some experience with low-level programming; Mathematica and Excel are tools that may be particularly useful.

Don and Steve's Projects

Truth, Lies, and Posets. (Don and Steve)

Description: Imagine that you are at Mathcamp. It's a fast-paced, exciting life, but there's one problem: some people are lying to you. Not everyone is a liar; some people always tell you the truth. However, you're having another problem: you don't know what your role is at camp. Are you a camper, a JC, a Mentor, or something else altogether?

Luckily, you can use these problems to attempt to solve each other, because they are connected: the entirety of Mathcamp exists on a poset, where whenever we have two people, A and B with $A \leq B$, person A always tells person B the truth, while person B always lies to person A. So, by interacting with those around you (who know their roles, but not about the existence of the poset), you can try to figure out both how different groups of people interact with each other, and where you fit into the hierarchy.

We'll start by placing strict assumptions on the structure of the poset, and seeing how much we can deduce from a fixed number of questions. Then, we'll slowly generalize the problem, increasing both the potential number of different posets, the number of allowed interactions, and the ways that incomparable people can interact.

Structure: 2–6 campers working to solve each new iteration of this problem; we'll give you the parameters, you'll work out how much can be solved within those parameters, and then repeat.

Expected Input: You can work as much or as little as you want on this project, and the amount you get out of it will likely be proportional to the amount of work you put in. However, it will be most rewarding to develop some amount of general theory of truthtelling-posets, which will require at least a few iterations of the parameter-solution process.

Expected Output: A poster outlining solutions to various special cases and the general approach to solving this kind of problem.

Difficulty:

Prerequisites: None

Jeff and Don's Projects

Food Homology. (Jeff and Don)

Description: A classical problem in mathematical cuisine asks: if a collection of foods are pairwise delicious, is there a a recipe containing all of them? Using the latest in modern topological and algebraic machinery, you will develop the tools to attack this problem.

Structure: 2–4 campers, meeting regularly. We'll give you some low-level topological and algebraic tools that we think are useful for this problem, and then set you loose.

Expected Input: 5 hours/week, mostly studying with tasting mixed in.

Expected Output: An understanding of how to solve the three foods problem, if not a solution.

Difficulty:

Prerequisites: Linear Algebra

Jeff's Projects

Brussel Sprouts on Surfaces. (Jeff)

Description: In the game Brussels Sprouts, you and a friend take turns building up a graph on a sheet of paper. Here are the rules:

- You begin with a bunch of small crosses, which have 4 free ends.
- Each player on their turn adds an edge connecting two free ends of the existing graph. The graph must remain planar when this move is made. They conclude their turn by adding a short tick mark across the new edge, creating 2 new free ends.

The game ends when a player can no longer move. Sounds like a great game, right?

Unfortunately, it is not a very good game, because it always ends in $5n - 2$ turns, where n is the number of starting crosses. Which makes it pretty dull.

Let's take this game and give it a twist. Let X be a topological surface. In X-sprouts, you take turns building up a graph on X. Suddenly, this game is no longer trivial! This project has two goals:

- What is a winning strategy for X-sprouts?
- If you know which player wins X sprouts for many different starting positions, can you figure out what space X is?

Structure: 5 hours/week, 2–3 students.

Expected Input: Students work together first building up example cases, developing notation for classifying games, and recognizing patterns in strategies for different starting positions.

Expected Output: A small goal would be a winning strategy for playing on the torus, Klein bottle and sphere. A secondary goal would be to identify a surface's genus and orientability from nim values for starting positions. An ambitious goal would be a winning strategy for every starting position on every surface.

Difficulty:

Prerequisites: Combinatorial Game Theory

The 2048calypse. (Jeff)

Description: Last year, a plague struck math departments across the United States. A addictive, habit-forming and destructive game called 2048 was released to the general public. Early estimates suggests that the whole of mathematics was set back by as much as a week.

It turns out we were lucky. In this project, you will research even more dangerous versions of this game. Is it possible to engineer a version where the player will never win? Are there ways to ensure that the player can make 1024, but never get to 2048? Your work today could save thousands of mathematicians in the future...

Structure: 2–4 campers who are willing to commit 3–5 hours a week designing devilish algorithms for 2048.

Expected Input: 3 hours/week

Expected Output: A version of 2048 that is impossible to beat. If possible, a version of 2048 which tempts the player into thinking they can win, but then defeats them at the last moment.

Difficulty:

Prerequisites: Ability to program a version of the game.

Yarnmobile Los Angeles. (Jeff)

Description: mmmMike: Hey Jeff! What did you do today in Los Angeles?

Jeff : Oh man! I tied a giant ball of yarn to the back of my car, and drove around Los Angeles. Here is where I went:

(1) Head southeast on W Temple St toward N Beaudry Ave
(2) Turn left onto N Grand Ave
(3) Turn left onto Alpine St
(4) Turn right onto the California 110 N ramp
(5) Merge onto CA–110 N
(6) Take the exit toward Dodger Stadium
(7) Turn left onto Stadium Way
(8) Turn left onto the California 110 S ramp
(9) Merge onto CA–110
(10) Take the Interstate 110 S exit toward 6th St/Wilshire Blvd/4th St/3rd St
(11) Keep right at the fork, follow signs for 3rd St/2nd St
(12) Turn right onto S Beaudry Ave
(13) Take the 1st right onto W 2nd St
(14) Turn left at the 2nd cross street onto S Figueroa St
(15) Turn right onto W 1st St
(16) Turn left onto N Broadway
(17) Turn right onto W Aliso St
(18) Turn left onto N Los Angeles St
(19) Turn right to merge onto US–101 N toward Hollywood/CA–110/I–110/Pasadena/Harbour Fwys
(20) Take the Echo Park Ave exit toward Glendale Blvd Continue on Bellevue Ave to W Temple St
(21) Sharp right onto Bellevue Ave
(22) Turn right onto E Edgeware Rd Turn left onto W Temple St

mmmmMike: Wait? Isn't that where you started?

Jeff: No! When I picked up the yarn, I noticed that it made a giant 6.8 mile long trefoil! The path that I drove is not isotopic to the constant path!

mmmmMike: . . . Whatever.

Jeff: I wonder what other knots I can make in this fashion? Maybe the campers can help me!

Structure: 2–3 campers who put in at least 5 hours a week to learn about knot invariant and apply them to freeway systems.

Expected Input: A couple of hours each week to learn about knot invariants, and try to generalize them to highway systems.

Expected Output: An invariant that provides useful information as to whether a knot appears in a freeway system or not. If possible, a knot that cannot be created in Los Angeles by driving along major highways.

Difficulty:

Prerequisites: Knot theory would be helpful, but is not required.

KEVIN, SARAH, ANGELA & GLORIA'S PROJECTS

Mathematical Crochet and Knitting. (Kevin, Sarah, Angela & Gloria)

Description: One of the best ways to visualize surfaces in three dimensions is to hold them in your hands and play with them. In this project, we'll make our own hyperbolic planes, Möbius strips, Klein bottles, Seifert surfaces, Lorenz manifolds, and more, all out of yarn! No previous crocheting or knitting experience is necessary.

Structure: We'll have introductory "how to crochet" and "how to knit" sessions; after that, you can work on projects pretty much wherever and whenever you feel like it! Extra instruction will be needed for Kat Bordhi's Möbius cast on, and for grafting. Technical help will be available during TAU.

Expected Input: Totally up to you. Expect a few hours to learn the basics if you've never crocheted or knitted before; after that, it's up to you.

Expected Output: Mathematical surfaces that you can hold and play with!

Difficulty:

Prerequisites: None

MARISA'S PROJECTS

The Graph Coloring Game. (Marisa)

Description: Alyssa wants to color the vertices of a planar graph with four colors so that adjacent vertices get different colors. If she picks up some crayons and starts without a plan, she might go through some false starts before she finds a good coloring: if she is unlucky, she might make a partial coloring that cannot be extended to a proper coloring of the whole graph.

Many hands make light work! Alyssa employs the services of her helpful friend Bob. They take turns coloring vertices, starting with Alyssa, and Bob agrees that he will not do stupid stuff (i.e., he will not violate the terms of the proper coloring). However, he is not being so excellent to Alyssa: his secret aim is to run them into a bad partial coloring so that they will not be able to complete the task. Is it possible for Alyssa to get her whole graph colored with four colors, in spite of Bob's lack of cooperation? If not, then how many additional colors are needed to guarantee that the graph can be successfully colored, no matter how devious Bob is in his choices?

This is the Graph Coloring Game. It was originally invented in an attempt to prove the Four-Color Theorem, and although it didn't achieve that goal, the game itself turns out to be interesting. It gives us questions like: What is the fewest number of colors allowing a guaranteed win for Alyssa on a particular type of graph, like a tree or a planar graph or a toroidal graph?

In this project, you'll explore Alyssa's strategies for winning the Graph Coloring Game and look for bounds on the "game chromatic number" of graphs.

Structure: This can be a solo or small group project. Plan on an initial meeting to get started, and about two check-ins with Marisa per week during TAU to discuss progress.

Expected Input: Very flexible. You can work on this for a few hours a week or a few hours a day.

Expected Output: Proportional to input! A poster would be a likely outcome. If you are very, very invested, you might prove something new and publishable. (But not likely during camp.)

Difficulty:

Prerequisites: You'll definitely need Intro Graph Theory. Also related (but not required) are Graph Colorings and Combinatorial Game Theory.

Mira's Projects

Hanabi AI. (Mira)

Description: Can we program a computer to play Hanabi? There are many possible approaches to this problem: you could build in rigid conventions and simply have the program compute lots of probabilities. You could take the "theory of mind" approach and have it try to figure out what the other players are thinking, using Church (the probabilistic computing language that Noah Goodman described in his colloquium). You could try to have it figure out the optimal conventions itself (though this sounds like it would be quite hard). You could have the program play with humans, several copies of itself, or other programs. You could even tell it to adopt different methods for playing humans (who have limited computational capacity) and for playing fellow AIs. The only limits in this project are your time, imagination and programming fu.

Structure: This project can take lots of students – different people or teams can work independently (and we can have their programs play each other)! We'll have an initial meeting to discuss possible approaches to the problem; after that, we'll only meet at your initiative – if you have questions, want to brainstorm some more ideas, or after you've made some interesting progress.

Expected Input: I suspect you won't be able to get anywhere interesting without putting in at least 5 hours per week. The ambitious versions of the project will require significantly more time.

Expected Output: A program that plays Hanabi!

Difficulty: 🌶🌶

Prerequisites: Good programming skills in some language

SVD and PCA. (Mira)

Description: The Mathcamp linear algebra course generally ends with the Spectral Theorem: every $n \times n$ symmetric has n orthonormal eigenvectors. But there's not enough time to convey to you just how important and powerful this result is, both within math and in a host of scientific applications.

Singular value decomposition (SVD) and Principal Component Analysis (PCA) are closely related techniques that build on the Spectral Theorem. They are used in a host of applications, from psychology to image processing to genetics (as well as within math itself). In this project, you will learn the math behind SVD/PCA and then choose one of several applications to focus on.

Structure: Several students can do this project, working in teams or as individuals on different applications. We'll meet a few times at the beginning so that I can teach you the math behind SVD. After that, you can choose an application to work on. I have a few in mind, and it shouldn't be hard to find more. The ones I'm most curious about involve analyzing data from surveys (say of your fellow campers) to find out the structure of people's preferences and/or personalities. But there are also lots of topics where the data you'd be working with can be simulated or found online.

Expected Input: A couple of hours of learning the math. After that, the simplest applications will take another few hours, whereas others could take sustained effort over the next three weeks.

Expected Output: Project fair poster with cool, easy-to-appreciate results.

Difficulty: 🌶🌶🌶

Prerequisites: The main prerequisite is the Spectral Theorem from Linear Algebra. Most applications also require some programming, but this can be pretty basic and you can learn as you go.

Where Are We? (Mira)

Description: In this project, we'll figure out our place in the universe the old-fashioned way, using celestial navigation and spherical trigonometry.

You may have noticed that in the Mathcamp library, we have an instrument called a sextant. A sextant is used to measure the angle between two distant objects, most often for the purposes of celestial navigation (finding one's position by the stars). Finding the latitude is easy. Finding the longitude is much harder – for several centuries, it was one of the great unsolved problems in sciene. You can't find longitude with just a sextant: you need either a consistently accurate clock or astronomical charts. Still, even in our age of GPS, most ships still carry a sextant on board and sailors still train to use it for emergencies.

Structure: This project can take 1-3 students. (We only have one sextant, so more people would be impractical.) At this point, I don't really know how to use a sextant (I just bought it a month ago), but it comes with an instruction manual, and there are lots of resources online, both on the use of a sextant and on the mathematics of longitude. I'm hoping we can figure it out together – or better yet, you can teach me.

Expected Input: I think it shouldn't take more than an hour or two to figure out how to get a reading using the sun and a clock. Without a clock, you need to use the stars and astronomical tables; this will take another several hours to sort out, especially if, like me, you're not that great at recognizing stars in the first place. You may also want to read up on the mathematics of longitude to better understand what you're doing.

Expected Output: Demonstrating your calculations at the project fair. Also, enhanced ability to survive a shipwreck.

Difficulty: 🌶🌶

Prerequisites: None

Misha's Projects

Graham–Rothschild Parameter Sets Theorem. (Misha)

Description: The Graham–Rothschild Parameter Sets Theorem is a far-reaching generalization of the Hales–Jewett theorem. There is a short proof of it that gives good upper bounds; unfortunately, the proof was written by logicians, so it's hard to understand.

Your mission, if you choose to accept it, is to read the proof, understand it, and write up your own explanation of how it works.

Structure: You can work alone or in a group. I will tell you everything I know about this theorem in a mini-crash course, then we will meet periodically to discuss your progress.

Expected Input: At least an hour a day, possibly more.

Expected Output: A write-up of the proof, written in a way that people can understand. This can be in any format you choose, but I recommend LaTeX. (Learning enough LaTeX to do this will be trivial compared to the actual work involved.)

Also you will understand one of the hardest results in Ramsey theory.

Difficulty: 🌶🌶🌶🌶

Prerequisites: My class on the Hales–Jewett theorem

Mathematica and Scientific Data. (Misha)

Description: Among other things, Mathematica grants easy access to lots of data, such as:

- Weather records going back decades from weather stations around the world.
- Mass, luminosity, and dozens of other properties for nearly 90 000 stars.
- The chemical structure of 44 096 different compounds.

Do you have questions that you think this kind of data might hold the answer to? If you can think of any (whether or not you know how to approach them, and whether or not you know how to use Mathematica) then come talk to me.

Structure: There can be multiple instances of this project (with different questions); I think any given instance is best with 1–3 people. I will get you started with using Mathematica; after this, I will set you loose.

Expected Input: Most questions can probably be answered in a few hours of work with Mathematica. Some may take longer.

Expected Output: An answer to your question (if you like, you can make a poster with graphs and such). Familiarity with Mathematica.

Difficulty:

Prerequisites: Interest in learning Mathematica; curiosity

Mathematica and Zome. (Misha)

Description: How many different tetrahedra can you make out of Zome? I think this question is too hard to solve by hand, but can be solved by brute force in Mathematica: the command

```
PolyhedronData["SmallRhombicosidodecahedron"]
```

will make a Zomeball, and you can go from there. Optionally, you can also learn to make 3D images of these things instead of touching actual Zome.

No prior knowledge of Mathematica is necessary.

Structure: This project can accommodate up to four campers, who will meet with me periodically during TAU.

Expected Input: Depending on the number of campers (more campers does not necessarily make programming easier) and their programming experience, this project could take anywhere between a few hours to around an hour of work a day. If we go quickly, there are further questions we can ask.

Expected Output: An answer to the above question, possibly with pictures and/or actual Zome constructions.

Difficulty:

Prerequisites: Interest in learning Mathematica

Misha and Pesto's Projects

"Problem Solving Through Problems" Solution Guide. (Misha and Pesto)

Description: *Problem Solving Through Problems* by Loren C. Larson is a great resource for olympiad competitors, problem solvers, and math enthusiasts. It covers a wide array of fundamental topics and techniques, and it reinforces each with numerous, well-chosen problems covering a range of difficulty levels. There is, however, an important downside: it has no solution manual. Let's fix that!

In this project, you will gain practice solving challenging olympiad problems at a level appropriate to your background, as well as experience in mathematical writing and LaTeX typesetting (no prior LaTeX experience required). The final product will be an invaluable addition to an already wonderful resource, to the benefit of many future problem solvers.

(This project was run in the past, and some of the problems have already been solved. Let's solve some more!)

Structure: You can work alone or in groups to solve problems from chapters of your choosing at difficulties of your choosing. You will also type up brief hints and solutions in LaTeX. We will meet periodically throughout this process to review your solutions, solve more challenging problems together, discuss strategies for write-ups, and polish the results. There is no upper bound to the number of participants in this project; the more the merrier!

Expected Input: Time spent on this project is flexible; whether you solve and contribute write-ups for five problems or five chapters is entirely up to you.

Expected Output: The tangible result of this project is a solution manual (or as much as possible) to a great problem-solving resource. Side-effects include bloated olympiad scores, heightened levels of mathematical confidence, and +5 LaTeX experience points.

Difficulty: 🌶🌶🌶

Prerequisites: Willingness to learn LaTeX

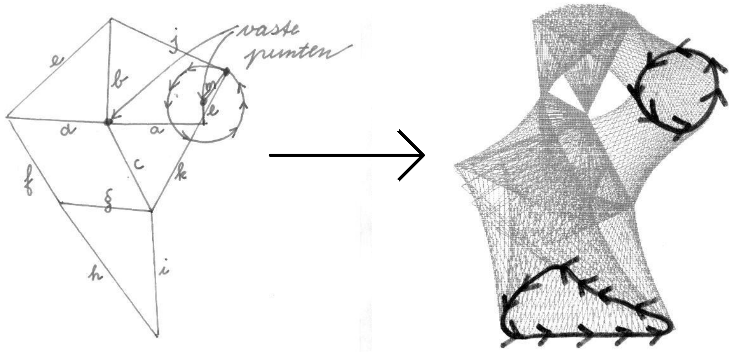
Paddy's Projects

Build a Strandbeest! (Paddy)

Description: This is a strandbeest!

It is a small multi-legged walking machine propelled by either the wind or a small motor. It is also incredibly cute (seriously, look them up on YouTube).

We're going to make one! In particular, we're going to both construct a strandbeest and explore the mathematics behind its design.



Students involved in this project will hopefully create a strandbeest of some level of functionality.

Structure: We'll meet a few times, get supplies, and create a strandbeest.

Expected Input: 12 hours in total.

Expected Output: A strandbeest! Also maybe some pretty pictures.

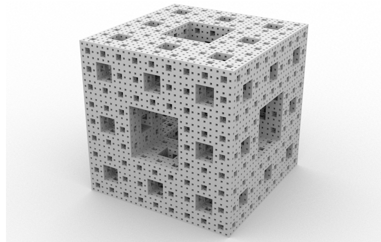
Difficulty: 🌶🌶

Prerequisites: None!

Menger Sponge Cake. (Paddy)

Description:

This is a Menger sponge:



This is a sponge cake:

Two years ago, we made a Menger Sponge Cake!

Well, kinda. For structural reasons, we opted to go with a more-solid cake form to help with stability. Unfortunately, this caused people to question our work! (But not our deliciousness.)

This year, with the aid of gingerbread Sierpinski carpets, we will fix this problem, and create an *actual* Menger Sponge Cake. Students involved in this project will explore the open question of how delicious cake can be when it has Hausdorff dimension $\frac{\log(20)}{\log(3)}$.

Structure: We'll bake things.

Expected Input: Depends on the number of iterations you want to go for, but probably about 8 hours of baking.

Expected Output: A cake!

Difficulty:

Prerequisites: None!

NP-Completeness and Latin Squares. (Paddy)

Description: A **Latin square** is a $n \times n$ array filled with the symbols $\{1, \dots n\}$, such that no rows or columns contain repeated symbols:

1	2	3
2	3	1
3	1	2

A **partial Latin square** is basically the same thing, except we let cells contain blank entries as well:

1		
	1	
		2

Determining whether some given set of Latin squares can be completed to a partial Latin square is the rare kind of problem that is NP-complete, open when specialized to almost any specific instance, and yet somehow an element of popular culture (see: Sudoku.)

In week 4, I'm running a class that shows the first of those properties: that completing an arbitrary Latin square is NP-complete. In my dissertation, I proved that completing a ϵ-sparse

Latin square — one that contains no more than ϵn filled entries in any row or column, and no more than ϵn cells containing any given symbol — can be done in polynomial time, for $\epsilon = \frac{1}{10^4}$.

So: somewhere between $\epsilon = 1$ (an arbitrary square) and $\epsilon = \frac{1}{10^4}$ (dissertation!), this problem switches from being "easy" to solve to being "hard" to solve.

I claim that it is not *too* difficult to improve 1 to some smaller constant, like 1/2. Campers involved in this project will either confirm or disprove this claim.

Structure: Biweekly meetings, through camp and possibly after camp ends.

Expected Input: As much time as campers can offer.

Expected Output: A resolution of an open problem that I've been interested in for a while.

Difficulty:

Prerequisites: An understanding of Latin squares; ideally, also understanding what P and NP are, though I could work without this.

Strongly Regular Graphs. (Paddy)

Description: A k**-regular** graph is simply a graph in which every vertex has degree k. This notion can be strengthened to that of a (n, k, λ, μ) **strongly regular graph**: a k-regular graph[1] on n vertices, such that

- any two adjacent vertices have precisely λ neighbors in common, while
- any two nonadjacent vertices have precisely μ neighbors in common.

Some examples: the pentagon is a $(5, 2, 0, 1)$, while the Petersen graph is a $(10, 3, 0, 1)$.

Being strongly regular is an incredibly restrictive graph property. In particular, consider the problem of fixing λ and μ ahead of time, and finding all of the strongly regular graphs (SRG's) with parameter set (n, k, λ, μ). For most sets of these parameters there are no such possible graphs, and for many others there are only finitely many; furthermore, in many of these cases we do not yet know how many such graphs exist! Famously, it is currently unknown how many graphs exist with parameter set $(n, k, 0, 1)$; we know that $(5, 2, 0, 1), (10, 3, 0, 1)$ and $(50, 7, 0, 1)$ exist, and that there may or may not be a $(3250, 57, 0, 1)$.

There is a list, maintained here, that matches up strongly regular graph parameters with known existence statuses:

http://www.win.tue.nl/ aeb/graphs/srg/srgtab.html

Students in this project will attempt to determine whether a given set of parameters on this list can be realized by a corresponding graph. The difficulty of this project will vary depending on the parameters chosen, and can range from a fairly simple/fun exploration of the overlap between linear algebra and graph theory to research on open problems in mathematics.

Structure: Biweekly meetings, both throughout camp and possibly after camp.

Expected Input: As much time as campers have and want to put into the project.

Expected Output: Either the discovery of a new graph, or the proof that a given set of parameters cannot be realized by any graph.

Difficulty:

Prerequisites: Linear algebra, graph theory.

The Smallest Natural Number Game. (Paddy)

Description: The smallest natural number game, in its purest form, is a Mathcamp game in the tradition of "Color or Country" or "The Penguin Name." It is an n-player game, and works as follows:

(1) Each player thinks of a natural number.

---

[1] For convenience's sake, we ask that these graphs are connected and not equal to a complete graph K_n, as such examples aren't very interesting.

(2) Everyone says their number out loud at the same time.
(3) The person who says the smallest number not named by anyone else wins!

It is a dumb and great game. Here is a variation on it:

(1) Play k matches in a row; say, 10^{10}.
(2) Keep score by giving someone a point for each match they win.
(3) Disallow anyone from naming whatever number they played on their last turn. (In other words, you can't simply say "0,0,0,0…" forever, but you **can** say "0,1,0,1,…")

What strategies should you adopt to get the most points possible? We will attempt to answer this question in this project.

Structure: Depends on the group of campers, but likely biweekly meetings, with a potential camp-wide tournament to test out strategies.

Expected Input: 4 hours/week.

Expected Output: An understanding of the smallest natural number game and its generalizations!

Difficulty: 🌶🌶

Prerequisites: None!

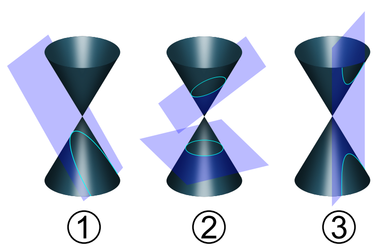
Paddy and Sachi's Projects

Sconic Sections. (Paddy, Sachi)

Description:

These are scones:

These are conic sections:



We're going to make sconic sections! Students involved in this project will get to explore whether hyperbolas are more delicious than ellipses, ponder whether the empty scone is really a scone at all, and ~~study the problem of how many scones can be tangent to five other scones~~ *eat delicious scones.*

Structure: We'll assemble molds and bake scones, possibly with delicious ganache.

Expected Input: 3–5 hours of baking.

Expected Output: Scones that look suspiciously like conic sections.

Difficulty: 🌶

Prerequisites: None.

Prerequisites: Nope!

Pesto's Projects

Graph Minors Research. (Pesto)

Description: In Pesto's undergraduate thesis, he made a conjecture that he couldn't solve, but that he thinks might be approachable with no more than an hour's worth of graph theory background teaching. The statement:

Conjecture 1. *For all positive integers k and all 2-connected simple graphs G with $|V(G)| \geq k+3$ and $\frac{|E(G)|-1}{|V(G)|-2} > \frac{3k-1}{k}$ there exists a 2-connected simple graph H such that $G \geq H$, $|V(H)| \geq k+2$, and $|E(H)| - 3|V(H)| \geq -6$, where $\geq$ is minor containment.*

Structure: Pesto'll teach the background at TAU one day, and meet every other day or so to talk about any ideas you've had.

Expected Input: At least an hour to learn enough to understand the problems.

Expected Output: Minimum (1 hour): a bit of understanding of graph theory. Medium (16 hours of thought, times or divided by 4): a solution to an easier version of the conjecture that Pesto solved. Maximum (even if you put as much time as possible in and do well, you might not get anywhere): prove (or disprove) the conjecture.

Difficulty: 

Prerequisites: None.

Models of Computation Stronger than P. (Pesto)

Description: Reading project about some topics that neither MCSP course covered, using Sipser's *Introduction to the Theory of Computation.*

Structure: Reading course: you read, and occasionally we meet to discuss what you've read.

Expected Input: Reading time: the expected amount you'll get out of it scales about linearly with the time you put in.

Expected Output: Some more MCSP knowledge

Difficulty: 

Prerequisites: Models of Computation as Strong as Programming (ask Pesto if you're not sure)

Ruthi's Projects

Commutative Algebra. (Ruthi)

Description: Like polynomials?Like ring theory? Think you might like ring theory? Think learning things by reading books or listening to lectures isn't as much fun as just doing a lot problems?

Commutative algebra is the study of sets in which you have both addition and multiplication behaving how you'd like them to — commutatively. It is critically important both for algebraic geometry and algebraic number theory, which it was essentially developed in concert with. If you like algebra and want to consider some fun abstract problems, this is a good project for you!

Structure: The funnest part of commutative algebra is doing the problems! I'll let you develop the theory and main theorems by doing the problems yourself, and then you'll come show me when you think you've solved them.

Expected Input: You'll go at your own pace, but if you want to learn it thoroughly, I think you'll need an hour a day on average.

Expected Output: A more thorough knowledge of commutative algebra!

Difficulty: 

Prerequisites: Ring theory is not necessarily a prereq. If you haven't taken it, I'll give you problems to help you learn some of the basics you'll need for the project.

Fun Frolicking with Finite Fields. (Ruthi)

Description: The classification of finite fields is a completely solved problem — but it's really fun to re-solve. In this project, I'll give you the questions that people ask to classify them (Can you get any size? What about subfields? Any other special properties?), and you'll work on trying to figure out the answers.

Structure: Meet with me occasionally and I'll give you a question to think about. If you get stuck, we'll talk about it some more.

Expected Input: You'll go at your own pace, but if you want to learn it thoroughly, I think you'll need an hour a day on average.

Expected Output: Understand finite fields completely!

Difficulty: 🌶🌶🌶

Prerequisites: Preferably ring theory and linear algebra. If you really want to do this project but don't have these, come talk to me.

Tim!'s Projects

A Peculiar Adversarial Channel. (Tim!)

Description: You want to send me a message! You encode the message as a string of n symbols (using some encoding scheme we've agreed upon ahead of time), where each symbol is 1, 2, or 3. I receive you message and decode it. There are 3^n distinct messages you can send me this way.

But now Eve starts intercepting your transmissions and modifies them. She can change any or all symbols if she wants to. Each symbol she has the option to change to the next (a 1 to a 2, a 2 to a 3, or a 3 to a 1), or to leave it the same. Given this interference, how can we design an encoding scheme that will allow us to communicate?

Frustratingly, no matter how big n is, and no matter how clever our encoding scheme is, I'll never know which message you meant to send. But, if we design our encoding scheme cleverly, I'll be able to narrow down the possibilities to two messages you could have sent — the message you really sent, and any one message that Eve wants me to think you sent.

Why can't we send messages in this scheme? How do we encode our messages so that I can always narrow it down to two possibilities? If there are m distinct messages that we want to be able to send, how big does n have to be to allow me to narrow it down to two (for specific values of m, and also the limit for large values of m)? What if the set of symbols is different, and the modifications that Eve is allowed to make are different?

These are all interesting problems that you can explore in this project. Many of these questions have attainable, interesting solutions, and some are very hard.

This problem is closely related to the Shannon Capacity of a Channel and Lovasz's Umbrella, which would be an interesting topic to read about.

Structure: You'll work on these problems, and we'll meet regularly to discuss what you've discovered.

Expected Input: You can spend as much or as little time as you want, but you will probably want to spend at least ten hours in total to really get into the problem and solve things.

Expected Output: A poster for the project fair.

Difficulty: 🌶🌶🌶

Prerequisites: None.

Do Mentors Know Calculus? (Tim!)

Description: While Prof. Tim Pennings (no relation) was playing with his dog Elvis at the beach, he was reminded of a classic calculus problem: If Tim throws a ball into the water, what is the fastest path that Elvis can take to the ball? Tim could tell "by the look in Elvis's eyes and his elevated excitement level" that the dog wanted to get to the ball as quickly as possible. So Tim decided to run an experiment, and he found that Elvis really did find the fastest route — the dog seemed to be solving this calculus problem. He wrote a paper about it — *Do Dogs Know Calculus?*.

For some reason, some people have been skeptical of this conclusion, and other tests have been carried out, resulting in the papers *Do Dogs Know Related Rates Rather Than Optimization?*, *Do Dogs Know Bifurcations?*, and *Dogs Don't Need Calculus*. But a lot of questions are left unanswered.

In this project, you'll form your own hypotheses about how we solve optimization problems on the fly in the real world, and you'll test them out. We don't have any dogs at camp to test them on, but we have the next best thing: Mathcamp staff.

Structure: You'll read some of the papers (they're short and amusing), learn about different ways of solving optimization problems, and then you'll design and carry out some experiments.

Expected Input: A few hours to read some papers and design and carry out an experiment, or longer if you want to develop a more complicated hypothesis

Expected Output: A poster for the project fair, reinforcement of the comparison between staff and animals with floppy ears.

Difficulty: 🌶

Prerequisites: Calculus (derivatives)