Contents

Alfonso's Projects

0.1. 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 Smith and Oppenheimer, and then you come back to me. 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 the fundamental group is, I will give you a crash tutorial on it first.

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: You need to know basic group theory. If you do not know what the fundamental group is, I will give you a crash tutorial on it first.

0.2. **Spoiled Projects in Combinatorial Game Theory.** (Alfonso)

Description: (These projects are intended for students who took Mira's or my CGT class in past Mathcamps. If you did not, see the project titled "Unspoiled projects..." instead.)

Here are various combinatorial games that are interesting to analyze.

- **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.

  You can solve this one fully by hand and. It has a nice pattern that requires some creativity.
- **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?

  This is an open problem, but there are various smaller questions that you can attack. It may help to do some coding to generate data. Some of the questions are open.
- **Variants of Nim.** (🌶–🌶🌶🌶🌶)
  You already know how to play nim. Here are three variants, from easy to very hard. For the harder problems, it will definitely be useful to generate some data with a computer in order to make and test conjectures.
  - *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.
- **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. You will benefit from a small amount of coding.

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: 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: 8 to 20 hours.

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: It depends (see each game)

Prerequisites: Having been in Mira's or my Combinatorial Game Theory class in a past Mathcamp. Alternatively, being familiar with the winning strategy for Nim and the MEX rule.

0.3. **The Coin Solitaire.** (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 presentation at the project fair.

Difficulty: 

Prerequisites: None

0.4. **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 – they all preferred to follow their "hunches", with sad results).

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? Lets 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).

0.5. **Unspoiled Projects in Combinatorial Games.** (Alfonso)

Description: (These projects are intended for students who did NOT take Mira's or my CGT class in past Mathcamps. If you did, see the project titled "Spoiled projects…" instead.)

Here are various combinatorial games that are interesting to analyze. The goal in all of them is to find a winning strategy.

- **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?
  You can fully solve and prove this one by hand.
- **Nim** 🌶🌶
  If you take on this project, do not let anybody tell you anything about the game of nim, and do not let anyone spoil you in any way. The game of nim has a beautiful and unexpected pattern as a solution, and I want you to have the pleasure of enjoying the discovery unspoiled.
  In this game, you and I play with various boxes of chocolates. In your turn, you may eat one or more chocolates, as long as they are from the same box. Then it is my turn. The player who eats the last chocolate wins. How do you proceed?
  You can solve this one entirely by hand, but it will require quite a bit of ingenuity and creativity.
- **Hysteresis Nim** 🌶🌶–🌶🌶🌶
  We play with a single box of chocolates. On his turn, the first player may eat any number of chocolates, at least one, but not the whole box. After that, each player has to eat a positive number of chocolates no greater than twice the number of chocolates eaten by the previous player. You win if you eat the last chocolate.
  After you solve this one, there are various follow-up questions. It will probably help if you can do some amount of coding to generate date before making conjectures.
- **The Rabbit and the Wolves** 🌶🌶–🌶🌶🌶
  A rabbit starts at the bottom of a long flight of stairs. Two wolves start on positions N and M. You and I play a game in which we move the rabbit and the wolves closer to each other. In your turn, you may move the rabbit any number of steps up (but not pass the wolves) or one of the wolves any number of steps down (but not pass the rabbit). The player who makes the three animal be on the same step wins.
  This game has an unexpected elegant pattern, with lots of combinatorics involved, and a difficult proof of the results. If you are a combinatorics junkie, this project is for you.

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: 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 of Euclid can be solved in 4 to 8 hours. For the rest, I would plan 8 to 20 hours.

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: It depends (see each game)

Prerequisites: None

0.6. **Resolving Square Dancing.** (Alfonso + Gloria + Ruthi)

Description: If you have been attending Gloria's "Dance by definition" sessions, you are familiar with 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 solving a Rubik's cube.

Here are various challenges:

- write good choreographies (they have to be smooth, flow well, and resolve),
- design a system to resolve from any configuration,
- be able to implement your resolving system on real time.

There is a computer program written by Bill Ackerman (available for free online; works on Windows and Linux) that allows you to try choreographies with virtual dancers. You may find this useful.

Structure: We will need to meet regularly to discuss your progress. At some point you will want to try it out on real dancers.

Expected Input: 10 to 20 hours

Expected Output: A poster at the Project Fair or a demo at the Talent Show.

Difficulty: 

Prerequisites: You need to have attended some of Gloria's "Dance-by-definition" sessions, or already be a square dancer.

Asilata's Projects

0.7. **Making Colourful String Pictures.** (Asilata)

Description: Root systems are some of the most beautiful and symmetric objects in mathematics, and you may have never even heard of them (and that's okay)!

In this project, we'll quickly get up to speed on what they are, and then model them using pretty string art. Let's show them off to Mathcamp! If you really get into it, you can branch out and make other mathematical string art too!

Structure: We will spend a short amount of time getting familiar with the mathematics, then brainstorm ways to make things using string. After that people can split off and work in groups.

Expected Input: Some independent study, group work with other campers to make several different root systems.

Expected Output: Gallery exhibition at the project fair.

Difficulty: 🌶🌶

Prerequisites: Linear algebra recommended

0.8. **Measuring the World.** (Asilata)

Description: Imagine you're a cavewoman-scientist. You really want to know All The Things about the world, but you don't have any tools! (Pretend you magically know what the SI units are, though.)

How would you measure the height of the dorm building, the depth of the swimming pool, the radius of the earth? Could you estimate the number of squirrels on the UPS campus, your own weight, or the latitude of Tacoma? How far can **you** get in measuring the world?

Structure: Self-driven by students. Ideally in group of 2–4. I will provide minimal guidance, but I can help with ideas.

Expected Input: Work with a group of people and come up with creative ways of solving various different estimation problems.

Expected Output: Possibly a presentation at the project fair.

Difficulty: 🌶

Prerequisites: None

0.9. **Choose Your Own Markov Chain Adventure.** (Asilata + Nina)

Description: A Markov chain is a simple way of modelling real-world phenomena, as well as lots of things you see at Mathcamp, such as board games! It is fun and illuminating to model some simple scenarios with the aid of a computer, to try and predict long-term behaviours.

In this project, you can choose to model a simple board game using a Markov chain. If you were in Nina's class, you maybe already worked on analyzing Risk. We could go further in that direction, or choose another process to model, like Monopoly or Can't Stop! or Chutes and Ladders, or come up with something completely different to study.

Structure: This is a coding project. I can work with you to choose a process and get started, but after that you'll also spend time on your own to work on code and analysis.

Expected Input: Flexible

Expected Output: An interactive presentation at the project fair

Difficulty: 🌶🌶🌶

Prerequisites: Intro to Markov chains helpful but not necessary, programming experience.

Chris's Projects

0.10. **Non-classical Constructions.** (Chris)

Description: If you took my non-classical constructions class in week 1, I should have left you with many questions. Almost all of them would make great problems. If you have any problems dear to your heart that you want to work on you can definitely come to me with this; additionally, here are some ideas of mine:

- In class we showed that $A \subset G$ where A is the set of numbers constructable with a marked ruler and G is the set of classically constructable numbers with only real conjugates. Show that $A = G$. This is very algebraic and technical.
- Show that every classically constructable regular n-gon is also constructable with a marked ruler. This is algebraic and builds onto $A = G$.
- Show the Mohr-Masceroni-Theorem: Every classical construction can be done with a compass alone. This is a geometric project similar to the proof of the Poncelet Steiner Theorem.
- Construct a regular pentagon just with a marked ruler in as little as possible steps.
- Investigate if our parallel construction with the marked ruler is minimal. This might be hard.
- Consider the following problem: Assume you are given a 1" ruler and a compass. Can you draw the line connecting two points that are 2.1" apart?

Structure: Individual/group study + meetings as needed

Expected Input: I expect you to think and work on one of the above problems as much as you have time and energy for.

Expected Output: Ideally, your work will produce proofs or constructions that will complement your coursework from week 1.

Difficulty: 🌶🌶

Prerequisites: Non-classical constructions

Don's Projects

0.11. **Abusing Category Theory for Fun and Profit.** (Don)

Description: Category theory was introduced in the 1940's, to study deep and difficult problems in algebraic topology. Since then, it has found use as a language of abstraction in fields as diverse as algebraic geometry and mathematical physics.

I don't want to talk about any of those. I want to talk about real-world things, forced into the language of category theory. What about the category of foods, where there is a morphism from one food to another if the former is an ingredient in the latter? What about the category of moral principles, where a morphism is a valid argument that one principle follows from another? What do categorical constructions or universal properties look like in these categories?

Structure: This will be a laid back project; we will encounter some very abstract concepts, but much of the hard work will be in thinking of interesting categories that aren't just posets. The best size is probably in the neighborhood of 3; enough to surprise each other with your ideas, but small enough to have comfortable group discussions.

Expected Input: We'll meet as often as you want, at least a couple times a week. I'll explain some questions we can ask of our categories, and you'll try to answer them, as well as come up with new interesting categories.

Expected Output: An understanding of universal properties, and an enhanced (some might say too enhanced) comfortability with abstraction.

Difficulty: 🌶🌶

Prerequisites: None, but if you already know category theory, this probably isn't the project for you.

0.12. **Liars, Truthtellers, and Moral Relativism.** (Don)

Description: In traditional Liars and Truthtellers puzzles, there are Knights, who always tell the truth, and Knaves, who always lie. That point of view is terribly outdated; people can't be broken into just two classifications; they identify with many different groups, and in some respects are each totally unique. On top of that, people don't act the same way no matter what situation they're in; you might say different things when talking to a friend than when talking to a stranger.

In this project, your task will be to create and classify puzzles in particular settings of relativistic truth telling.

Structure: This is a very open ended problem. There's a lot of source material on the old style of puzzles to reference, but you'll be building up this theory mostly from scratch.

Expected Input: You'll need to put in several hours a week at a minimum in order to really say anything interesting about this problem; to give a full account might take even more time.

Expected Output: An analysis of this problem, which might be too large to fit on a poster, but probably won't be too large for a presentation (or a crazy live-action demonstration).

Difficulty: 🌶🌶🌶🌶

Prerequisites: None

0.13. **Marco Polo on Graphs.** (Don)

Description: Marco Polo is a classic children's game, in which one player must close their eyes and try to catch any other player; in order to get information about the locations of the other players, they may shout "Marco!" to which the other players must respond "Polo!"

Actual Marco Polo is very hard to analyze. However, if we pretend that players all live on the vertices of graphs, that moves alternate between the pursuer and the pursued, and that on their turn, the pursuer can chose to either traverse one edge, or shout "marco" to hear the direction in which his prey lies, we can make some real progress.

Structure: This will work best as a small group project, from 1–3, meeting and working through cases. I have a number of small variations to ask about the problem. I don't know all the answers, but the ones I do know are already pretty cool.

Expected Input: Several hours a week, and a willing to work through cases, in order to come up with hypotheses you can then try to prove.

Expected Output: A full explanation of generalized Marco Polo games on graphs — and perhaps a crazy, live-action event.

Difficulty: 🌶🌶

Prerequisites: None

0.14. **Finite Topological Spaces.** (Don + Yuval)

Description: Most topological spaces that mathematicians study are infinite, even uncountable. This might lead you to think that only infinite spaces are interesting. In fact, this could not be farther from the truth! In this project, well learn as much as we can about finite topological spaces. Well start by looking at examples of spaces with very small numbers of points to try and see what finite spaces can look like. Then well use these examples to figure out how to characterize all finite spaces (and continuous maps between them) with axioms that are much simpler than the general axioms for topological spaces. If you get far enough with this, we might even be able to learn a little about how finite spaces relate to algebraic topology.

Structure: You'll be doing most of the work; we'll guide you in productive directions and suggest ideas and definitions to think about. This project probably works best with two campers bouncing ideas off each other, although one camper can certainly make significant progress.

Expected Input: You should be ready to put in some amount of work every day; finite topological spaces are weird, and you'll be building your intuition for them from scratch.

Expected Output: As good an understanding of the behavior of finite spaces as we can get, and a poster about this if you want to make one.

Difficulty: 🌶🌶🌶🌶

Prerequisites: Point-set Topology

Gloria's Projects

0.15. **Mathematical Crochet and Knitting.** (Gloria + Nancy + Angela)

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 or felt or fabric! No previous crocheting or knitting experience is necessary.

Structure: We'll have introductory "how to" 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 done any of these things before; after that, it's up to you.

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

Difficulty: 🌶

Prerequisites: None

J-Lo's Projects

0.16. **The Flip Flop Line.** (J-Lo)

Description: A countable number of people line up along the integers, and the person sitting at 0 starts a game of Flip Flop. Will everyone get a turn? Infinitely many turns? About how long will the n^{th} person have to wait before his/her first turn?

There are a lot of questions like the above that can be asked about this game. There is also a lot of room for exploration if you consider altering the rules slightly. Most ambitiously, can we find a modified version of Flip Flop that acts as a universal Turing machine while still resembling the original game?

(question set-up inspired by Evan Wu)

Structure: We will meet at least once a week to brainstorm questions to explore and to assign tasks — either collecting data (if you know basic programming) or proving results.

Expected Input: Minimum two hours per week

Expected Output: A poster presenting a collection of results

Difficulty: 🌶

Prerequisites: experience playing Flip Flop is preferred

Jalex's Projects

0.17. **Russian BS AI Tournament.** (Jalex)

Description: Russian BS is a competitive game with a central mechanic of deception. In this project, we'll explore the evolution of the Russian BS metagame in a population where all players have perfect memory and access to (near-perfect) randomness. In other words, we'll teach computers to lie effectively!

We'll program computers to play the game against each other, and experiment with lots of different strategies. We might hope to answer some questions relevant to actual human play:

- How powerful is perfect memory?
- How does your strategy change if you're optimizing for probability of first place versus expected value of place?
- When is it good pass up a guaranteed call?

The rules of the game are as follows:

A deck of standard playing cards is dealt among n players. One played begins a round by placing some cards face-down on the table and claims they are all of a particular rank. The next player has three options:

- Place a positive number of cards facedown on the table and claim they are of the same rank as the previous player.
- Call BS on the previous player.
- Call Believe on the previous player.

If cards are played, the next player takes a turn with the same three options. If a call is made, the cards played by the previous player are revealed, (they become public information) and then one of three things happens:

- If the call is "BS" and the cards played are not all of the same rank as was claimed, the previous player takes all cards on the table into their hand. The current player then starts a new round.
- If the call is "Believe" and the cards played are all of the same rank as was claimed, all cards on the table are removed from play permanently. The current player then starts a new round.
- Otherwise, the current player takes all cards on the table into their hand. The next player starts a new round.

A player wins when a round ends and they have no cards in hand. Play continues until all but one player has won, and players are ranked by the order they won in.

Structure: 2+ campers, so that we have multiple strategies to play against each other. I'll get you started with an engine to play strategies against each other, and then I'll be available for programming help and game theory ideas.

Expected Input: 4–20 hours of designing, implementing, and testing strategies

Expected Output: Lots of little lying AIs, and a deeper understanding of a fun game

Difficulty: 🌶🌶

Prerequisites: Some programming knowledge

Jeff's Projects

0.18. **Build a Radio!** (Jeff)

Description: One of the oldest electric technologies was the radio. An AM (amplitude modulated) radio station transmits a function $f(t)$ on frequency ξ by sending the signal

$$\phi(t) = f(t)\sin(\xi t)$$

into the air. Someone who receives this signal then takes $\phi(t)$, and (because they're trying to listen to (ξ)-frequency station) divides by $\sin(\xi t)$ to recover $\phi(t)$.
But ξ is not the only radio station that plays music. There is also station ζ, which transmits $\psi(t) = g(t)\sin(\zeta t)$. So, when someone listening in the radio station wants to listen to station ξ, the actually receive

$$\phi(t) + \psi(t).$$

How can they recover the original signal. Once we develope a mathematical method for doing this, we are going to build an electronic circuit which actually implements this mathematical algorithm, and listen to the airwaves.
Structure:

Expected Input: Students will work with advisor to learn mathematics behind band-pass filters, then we will build a band pass filter and amplifying circuit.

Expected Output: A radio!

Difficulty: 

Prerequisites: You should know Ohms Law. Calculus.

0.19. **Embedded Graphs.** (Jeff)

Description: Have you ever seen those puzzles with two horseshoes and a ring, where you are supposed to pull the ring off the horseshoes? Topologically the problem is solvable — take the rings and deform them so they slip off.
How do we know if a disentanglement puzzle has a topological solution? One way to study these problems would be to use knot theory — however, this limits us to the puzzles which are just made of various circles linked together. One can generalize invariants of knot theory to study embedded graphs, which describe all kinds of disentanglement puzzles.
Structure:

Expected Input: Weekly meetings with advisor to update on progress. 4–5 hours a week.

Expected Output: Adapting a knot invariant of your choice to work for embedded graphs.

Difficulty: 

Prerequisites: Knot Theory

0.20. **Games with Graphs.** (Jeff)

Description: The game of Brussel Sprouts is a game involving topological graphs. The playing field is a sheet of paper, with several x's drawn on it. On their turn, each player may take a point and connect it to another point by an edge that does not cross the other edges. They then add a little slash to that edge, creating two new points. This process slowly builds up a planar graph, until one of the two players is unable to make a move and loses.
This game depends not only on the graph created, but on the topology of the drawing of this graph. Come see me during the project fair if you're interested in playing this and other examples of games with graphs!
Structure:

Expected Input: Students will work in a group mostly without supervision. 5–10 hours a week.

Expected Output: A complete solution to the game of Brussel Sprouts, and partial solutions for the game of Soy Sprouts.

Difficulty: 

Prerequisites: Graph Theory

0.21. **Lines and Knots.** (Jeff)

Description: Ok, pretty obvious that every knot has a line that intersects it. In fact, it is also pretty clear that every knot has a line that intersects it at 2 points. Can you show that every knot (which is not the unknot) has a line that intersects it at 3 points? What about 4 points? What about 5 points?
Structure:

Expected Input: Work several hours a week towards finding trisecants or quadresecants. If you get stuck, or need ideas, meet up with advisor during TAU.

Expected Output: Show the existence of trisecants

Difficulty: 🌶🌶🌶

Prerequisites: Knot Theory or Point-Set Topology or Fundamental Group

0.22. **Zoo of Topological Counterexamples.** (Jeff)

Description: Here are some different definitions for topological connectedness:

> Connected, Path Connected, Arc Connected, Locally Connected, Hyperconnected, Uniformly Connected, n-Connected

Each type of connectedness is subtly different. Take a safari to the wild lands of topological spaces to catalogue examples that satisfy these unique traits, and many others. Then bring them back home and build a "map" for a "topological zoo", which exhibits the finest topological spaces you were able to find. For example, the Path Connected exhibit should be contained in the Connected exhibit of the zoo, but there should be at least one animal in the connected exhibit which is not in the path connected exhibit.

Structure: Students will be given a set of properties of topological spaces, and asked to find topological spaces that show that classify the relations between the properties.

Expected Input: 5–10 hours a week of independent work, meeting with advisor once a week.

Expected Output: Make a zoo for a map.

Difficulty: 🌶🌶🌶🌶

Prerequisites: Point-Set Topology

Julian's Projects

0.23. **The Tower of Hanoi and Friends.** (Julian + Jalex)

Description: The Tower of Hanoi is a well-known puzzle, and now sits in the games cupboard. There are many fascinating questions which can be asked about it or variants of it. I have a handout of suggestions including a variety of mathematical problems and also some psychological experiments based on the Tower of Hanoi and similar puzzles; this will be available at the Project Fair.

Structure: Depending on the specific problem investigated, it might be an individual or small group project; it could involve programming if that is wanted; the psychological ones will involve getting people (other campers?) to try puzzles.

Expected Input: Depending on the project chosen, anywhere from 4 hours upwards.

Expected Output: Depending on the project chosen, could be a poster, a game, a computer program.

Difficulty: 🌶–🌶🌶🌶🌶

Prerequisites: None

Mark's Projects

0.24. **Knight's Tours.** (Mark)

Description: Given a rectangular "chessboard" (of m by n squares), for what values of m and n can a knight traverse all the squares of the board (using legal moves) without ever visiting a square twice? (If you don't know how a knight moves in chess, ask.) For what values of m and n can this be done in such a way that the knight can leap back to the starting square from the ending square to create a "closed" knight's tour? In this project you'll be investigating these questions, and possibly (if you are dedicated and ingenious) answering them completely, or almost completely.

Structure: We'll meet to get you started, and then it's largely up to you.

Expected Input: At least several hours a week, but up to you. I can give hints and will be available during TAU, but I won't come after you if you don't persevere.

Expected Output: A poster for the project fair.

Difficulty: 🌶🌶

Prerequisites: None, although you should be comfortable with mathematical induction, and you should not mind doing a lot of careful experimentation — if you dislike looking at many different cases, this is not the project for you.

0.25. **Period of the Fibonacci Sequence Modulo n.** (Mark)

Description: The Fibonacci sequence modulo 8 looks like

$$1, 1, 2, 3, 5, 0, 5, 5, 2, 7, 1, 0, 1, 1, \dots\ ;$$

as you can see, it is periodic with period 12. In this project you'll investigate the period of the Fibonacci sequence modulo n as a function $f(n)$; we just saw that $f(8) = 12$. This should lead to some ideas from number theory, as well as to an open problem (that is somewhat notorious and probably unreasonably difficult).

Structure: This would probably work best with 2 or 3 campers. Although I will be available for help during TAU and I'll be willing to give hints, I will not hunt you down if you don't keep working actively on the project, so you should only sign up for this one if you are highly motivated.

Expected Input: Up to you. In order to make serious progress, you should probably spend at least three or four hours per week, but it depends a bit on how quick you are.

Early on in the project there will be some computer/programmable calculator work so you get enough data to make conjectures. Once you have a table of enough values for the function, the rest of your work will be by hand.

Expected Output: A poster for the project fair.

Difficulty: 🌶🌶–🌶🌶🌶, depending on your background

Prerequisites: A bit of number theory will be helpful, but as long as you know modular arithmetic you can work on the project and pick up the number theory as you go along. On the other hand, if you have had a serious college course on number theory (through the law of quadratic reciprocity) it is possible that the project won't be as challenging as you might like. It would be good if at least one participant has a little programming experience — it doesn't really matter in what language or environment, as long as you know that you can get results on this campus.

Mira's Projects

0.26. **Breaking Codes, Turing Style.** (Mira)

Description: There is a website called the ArXiv on which mathematicians upload all their papers (since getting a paper reviewed and published in a journal can take a long time). On May 15, 2015, a paper appeared on this website called *The Applications of Probability to Cryptography.* The author of the paper was Alan Turing.

No, Turing is not back from the dead (which is a shame, since he'd find our world fascinating if he were: internet! artificial intelligence! gay marriage!). This is a paper that Turing wrote during WWII, which has only recently been declassified. It describes the application of Bayesian inference (a statistical approach that is becoming increasingly important in AI and machine learning) to decoding Vigenere codes (which are similar to, though much simpler than, the Enigma code). None of the math in this paper is very complicated, but it feels remarkably modern. It's also very practical, containing actual examples of how one might try to break codes.

In this project, we will try to implement Turing's ideas – perhaps on the computer or perhaps even by hand (which is more or less how they did it during WWII). In the process, you will learn the basics of Bayesian statistics and gain an appreciation for how hard some tasks were before computers — and how much could nonetheless be accomplished.

Structure: This project can be done either individually or in a group. First you'll need to read and understand Turing's paper; I can help you out as much as you need. Then you'll decide which parts of the paper you want to implement and how much programming you want to do. And then you'll do it!

Expected Input: A few hours a week.

Expected Output: A working system for deciphering Vigenere codes based on Turing's ideas; a better understanding of probability and Bayesian inference.

Difficulty: 🌶🌶

Prerequisites: None — you can learn all the probability you need along the way. It would be good, though not necessary, if we had one or more people who can program.

0.27. **The Mathematics of Conectaballs.** (Mira)

Description: You may have noticed another mathematical construction toy next to the Zometool this year: Conectaballs. A friend of mine invented them and is now selling them on Kickstarter. He gave me a set, together with a math problem: prove that any graph that has edges of equal length, with three edges meeting at vertex, can be made out of Conectaballs. In return, I promised him I would try to get Mathcampers to come up with more mathematical problems (and solutions) involving Conectaballs. And that's why I'm running this project!

Structure: Any number of people can work on this, in groups or individually. First you solve my friend's Conectaball problem. Then you play with Conectaballs and come up with more problems and interesting things to build. I'm happy to discuss any ideas you have, but I don't know any more about this than you do!

Expected Input: Anywhere from a few hours to all your free time.

Expected Output: Lots of cool stuff made out of Conectaballs; making Mira's friend happy and probably getting your math problem/puzzle posted on the Conectaballs website.

Difficulty: 🌶

Prerequisites: None

0.28. **Zome: Bigger and Better.** (Mira)

Description: On Day 0 of Mathcamp, we built the 3D shadow of a regular 4D polytope (the 120-cell) using Zome. Do you want to build something even bigger (and maybe understand it a little better in the process)? We can build the projection of one of the Archimedean 4D polytopes. It will be huge. It will be awesome.

Structure: We figure out which structure we want to build. We make sure we have enough Zome for it and order more if we don't. We build it.

Expected Input: Playing with Zome.

Expected Output: A colorful shadow of an intricate 4D structure, which everyone will want to take pictures of.

Difficulty:

Prerequisites: None

0.29. **Sum and Product Knowledge Puzzles.** (Mira + Don)

Description: Here is a certain kind of logic puzzle: I'm thinking of two positive integers. I tell my friend Tim! their product, and my friend Adina their sum. I then alternate asking Tim! and Adina whether they know my numbers. Amazingly, after a number of back and forth questions, this can result in my friends figuring out my numbers.

Will this process always end? When will it end? What can an outside observer deduce about my numbers from just hearing the questions and answers? What happens when we start asking, not just about what they know, but about what they know of what the other person knows? Will we go mad thinking about this?

To answer these questions and more, join this project.

Structure: Any size group should be able to make progress; you can work together, or apart, trying to find the answers.

Expected Input: The amount of time you spend is flexible; you should work on it long enough to find a result that you find interesting!

Expected Output: This could result in a poster, and an analysis of these kinds of puzzles

Difficulty:

Prerequisites: None

0.30. **Make Your Own Hydras.** (Mira + Susan)

Description: The hydras in Susan's colloquium were wimpy! Take a look at `http://www.madore.org/~david/math/hydra.xhtml` — these hydras are much scarier. Sure, they're still ordinals, but they're much bigger ordinals. Actually, we haven't exactly figured out what ordinals they are — that's your job in this project. And once you do that, you can try to make even scarier hydras of your own and defeat them too.

Structure: We'll help you learn more about ordinals, so that you're better equipped to fight the hydras. Then you'll go int battle on your own, but we'll be available to comment from the sidelines whenever you want.

Expected Input: Flexible — maybe a few hours a week?

Expected Output: Probably a poster with proofs and hydras.

Difficulty:

Prerequisites: Courage in the face of mythical monsters and infinity.

Misha's Projects

0.31. **Graham's Number.** (Misha)

Description: Graham's number is the (famously enormous) upper bound to a combinatorial problem about coloring segments drawn between vertices of an n-dimensional cube. The lower bound started out at 6; it's been improved to 11 in 2003, and to 13 in 2008 by computerized search. (The upper bound is still larger than the number of atoms in the universe.)

Let's see how good a lower bound we can get. The existing results did not use any sophisticated methods, so it might be possible to do better than them, which would be very exciting; I think it would not be too hard to get a lower bound in the range 10–12.

Structure: After an initial meeting to get you up to speed on the problem, the rest is up to you.

Expected Input: I do not think it would go anywhere without at least 1–2 hours of work a week. Having a laptop that can work while you're not is helpful.

Expected Output: A coloring of all segments between vertices of an n-dimensional cube that does not contain monochromatic planes. The value of n will vary.

Difficulty:

Prerequisites: Some programming knowledge required

0.32. **Graham's Parallel Number.** (Misha)

Description: Graham's number is the (famously enormous) upper bound to a combinatorial problem about coloring segments drawn between vertices of an n-dimensional cube. The lower bound started out at 6; it's been improved to 11 in 2003, and to 13 in 2008 by computerized search. (The upper bound is still larger than the number of atoms in the universe.)

An interesting lemma in the proof of an upper bound to Graham's number is the following result: if you require that parallel segments receive the same color, then the correct answer to the problem is exactly 6. I know of no proof of this that does not use a computer.

At the very least, you will replicate the computer proof. You can then think about how to solve the problem without a computer (possibly for larger values of 6).

Structure: When working on the computer proof, I'll be very helpful.

When working on the real proof, I will be less helpful. I have ideas for approaches you can take, but I will avoid suggesting them until you think about the problem for a while first, to avoid influencing what you come up with.

Expected Input: Replicating the computer proof may only take 2–4 hours. Thinking about the problem may take arbitrary amounts of time.

Expected Output: You will learn stuff about Ramsey theory. It's very unlikely that you will prove 6, but you may possibly prove some slightly weaker bound such as 6 trillion.

Difficulty:

Prerequisites: Some knowledge of Mathematica or something similar helpful, but I can catch you up to speed.

0.33. **Many Creased Squares of Paper.** (Misha)

Description: (That is, modular origami)

The Soma cube is a seven-piece puzzle that can be assembled into a 3x3x3 cube. You can make one out of 122 squares of paper.

You can also make exciting polyhedra! My favorite is the snub cube, because you can't make it out of Zome, which made me very sad when I found out. But you can make one out of 60 sheets of paper.

There are other things, some easier and some harder; we can choose which to make depending on interest.

Structure: Folding 122 of the same thing is boring alone, so this project is best done in groups (the more, the merrier).

Expected Input: In a decent-sized group, we can probably get a single thing done in an evening. If there is further interest, we can get more things done in more evenings.

Expected Output: one or more 3D origami models.

Difficulty: 🌶

Prerequisites: None.

Munmun's Projects

0.34. **Generatingfunctionology.** (Munmun + Kevin)

Description: Generatingfunctionology by Herbert S. Wilf is a wonderfully written book which focuses on ordinary and exponential generating functions. If you're interested in learning more after Mark's class, or had some background with generating functions already, Wilf's book is a wonderful, well-written way to go into even more depth. One potential focus for our reading is the following question: given the number of connected, labelled graphs of at most n vertices, how can we develop a theory of counting and relabelling that will allow us to piece these connected parts into an arbitrary labelled graph of size n? We'll reach an understanding of the framework to answer this question more generally.

Structure: Read chapters of Wilf's online book, *generatingfunctionology*, on your own, we'll work through examples in meetings, and working through a handful of problems. Exactly which chapters we focus on will depend on interest.

Expected Input: Flexible, depending on interest. Some number of hours split between reading, doing problems on your own, and discussion. If you're interested, talk to Munmun and Kevin!

Expected Output: A stronger understanding of the beauty and utility of generating functions.

Difficulty: 🌶🌶🌶

Prerequisites: Some familiarity with ordinary generating functions and with Taylor expansions for functions such as $f(x) = e^x$ and $g(x) = sin(x)$.

Nancy's Projects

0.35. **Embedding Regular Polytopes in Integer Lattices.** (Nancy)

Description: Regular Polytopes are the generalizations of regular polygons and polyhedra in higher dimensions. There is a beautiful classification of the regular polytopes into collections of infinite families and a few exceptional shapes. Consider, the regular polygons the square and equilateral triangle. The square can be easily drawn in the plane with corners at points with integer coordinates, i.e. embedded in an integer lattice in $\mathbb{R}^2$. An equilateral triangle cannot be embedded in an integer lattice of the plane, but can in an integer lattice of $\mathbb{R}^3$. In this project, the camper will learn the classification of regular polytopes and for each polytope, find the minimum dimension (if it exists!) needed to embed that polytope into an integer lattice of $\mathbb{R}^n$.

Structure: I will meet with you as needed, up to three times a week.

Expected Input: You will do most of the work on their own, with some guidance.

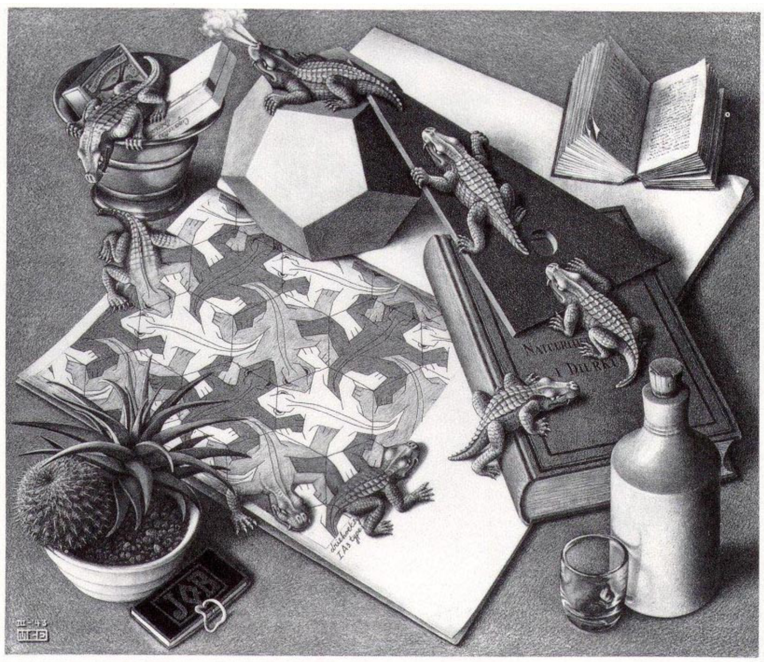
Expected Output: A short paper with a complete classification of dimension.

Difficulty: 🌶🌶🌶

Prerequisites: None.

0.36. **Escher Inspirations.** (Nancy)

Description: Escher is famous for his amazing optical illusions, tessellations and hyperbolic sense of space. Let's make art in an Escher style, eg.



Structure:

Expected Input:

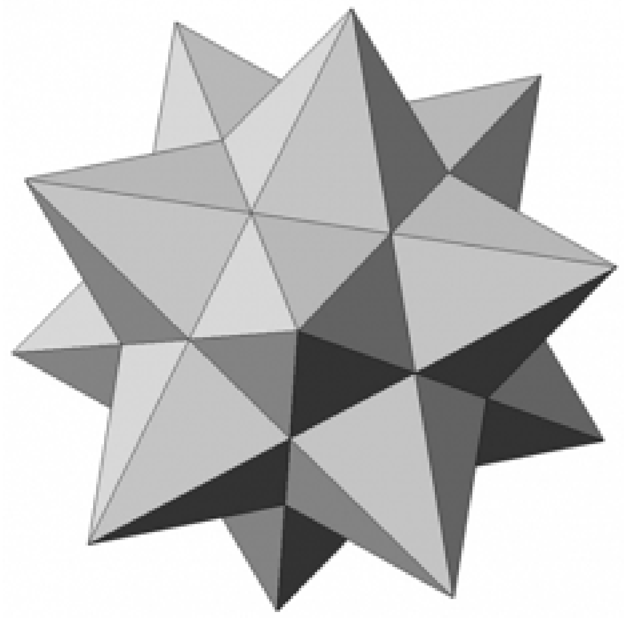
Expected Output: Art!

Difficulty: 🌶

Prerequisites: None.

0.37. **Polyhedral Sculptures.** (Nancy)

Description: Polyhedra are beautiful, relatively easy to make, and form an interesting canvas for creation.



Some ideas for sculptures are:

- Let's make our own paper and then create paper polyhedra. Perhaps coat the paper with something sturdy to help keeps its shape.
- Sew polyhedra out of fabric
- Make polyhedra out of papier mâché and paint the surface.

Structure:

Expected Input:

Expected Output: Polyhedra

Difficulty: 🌶

Prerequisites: None.

0.38. **Topological Jigsaw Puzzles.** (Nancy)

Description: I have a book full of instructions on how to make to make jigsaw puzzles. Let's make some of our own!

Structure: Let's make stuff whenever we have the time.
Expected Input:

Expected Output:

Difficulty: 🌶

Prerequisites: None.

Pesto's Projects

0.39. **Computer Science Reading.** (Pesto)

Description: Intro complexity, cryptography, and P vs NP not enough for you? Read a theoretical CS book, helped by Pesto.

Structure: Read a book and talk about it with Pesto at TAU.

Expected Input: Reading.

Expected Output: CS knowledge.

Difficulty:

Prerequisites: None.

0.40. **Graph Minors Research.** (Pesto)

Description: In Pestos undergraduate thesis, he made a conjecture that he couldnt solve, but that he thinks might be approachable with no more than an hours 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 $\frac{|E(G)|1}{|V(G)|2} > \frac{3k1}{k}$ there exists a 2-connected simple graph H such that $G \geq H$, $|V(H)| \geq k+3$, and $|E(H)|3|V(H)| \geq 6$, where $\geq$ is minor containment.*

Structure: Pestoll teach the background at TAU one day, and meet every other day or so to talk about any ideas youve 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 (24 hours of thought, times or divided by 2): a solution to an easier version of the conjecture (that's already a theorem).

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

0.41. **The Worst Election Ever.** (Pesto)

Description: You can have a plurality-voting election in which one candidate is more popular than each other candidate, but doesn't win. In "Instant Runoff Voting" (used in, e.g., Australia), making your ranking of a candidate higher can make them lose. These and similar paradoxes are usually illustrated by sample elections in which they occur: for instance, if

- 48 % of voters prefer Gore to Nader to Bush,
- 49 % prefer Bush to Gore to Nader, and
- 3 % prefer Nader to Gore to Bush,

then Gore would beat Bush and Gore would beat Nader, but Gore loses the 3-way election.

Can we make *one* such example election for which every one of the most famous paradoxes would occur in all of the most famous voting systems vulnerable to it, or are there sets of (paradox, voting system) pairs that cant all happen simultaneously? Is there a systematic way to combine examples of some of the paradoxes, so we dont have to create one big counterexample all at once?

Structure: Meetings at TAU: daily first, then less frequently.

Expected Input: The more time and effort you put in, the worse a final product you can get.

Expected Output: An exquisitely awful election, and perhaps some theorems.

Difficulty:

Prerequisites: If you haven't already seen any voting theory, we'll do it at the start of the project.

Ruthi's Projects

0.42. **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: Some commutative algebra in your brain!

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.

0.43. **Reading Tate and Silverman.** (Ruthi)

Description: Elliptic curves are some of the important objects in number theory, not to mention deeply beautiful. They are used to solve problems all over number theory, and were integral to the solution to Fermat's Last Theorem[1]. Silverman and Tate's *Rational Points on Elliptic Curves* is a lovely introduction to the topic, and this would be a reading course with this book.

Structure: We alternate between you reading a bit and asking me questions about what you're reading. I'm also happy to motivate the book more with examples and context for how what you're doing is relevant.

Expected Input: Reading 6+ hours/week, meeting 2 hours/week

Expected Output: A beginning to understanding elliptic curves over the rationals.

Difficulty: 🌶🌶🌶

Prerequisites: Ring theory.

Sachi's Projects

0.44. **Arduino.** (Sachi + Kevin)

Description: An arduino is a tiny computer that controls electronic circuits. We can connect it to LEDs, sensors, buzzers, motors, and more. We can build robots that draw, musical instruments triggered by light, and filter sound input to make you sound like your favorite mentor.

Structure: Students will work in groups of two to four. Projects may focus on any or all of programming, electronics, and design. All projects will involve elements of all three, but no previous experience in any of them is required.

Expected Input: Regular project meetings; more ambitious projects will require more time.

Expected Output: Something awesome powered by an arduino!

Difficulty: 🌶🌶

Prerequisites: Excitement for hands-on tinkering.

[1]To be clear: you will not end this project understanding the proof of Fermat's Last Theorem.

Sam's Projects

0.45. **History of Math!** (Sam)

Description: This is a pretty flexible project, and we can tailor it to what you want. The main idea is to find something you're interested in — a specific mathematician, the development of an area of mathematics, the development of a specific result, the interplay between math and something else — and learn about it! Then we'll set up some formal or informal presentation about it! For a bonus, you could also try to learn some math from a famous historical source (one of Turing's papers, Cauchy's Cours d'Analyse, etc).

Structure: Highly variant — let's talk about how everything would work.

Expected Input: Come with ideas about what you want to learn about, and willingness to learn about those things.

Expected Output: Also variant!

Difficulty: 🌶🌶

Prerequisites: Enthusiasm!

0.46. **Statistical Modelling.** (Sam)

Description: Let's analyze some real data sets that you care about! For example:

- Can we predict how well someone will do in chess-sprinting based on their chess rating or their 100M sprint time? What else might help?
- What affects how much it costs to have certain celebrities play a concert at your house?

Structure: Analysis will use R! We'll meet when you need to, and will discuss specific questions and analysis procedures that might help. You'll then do most of the analysis on your own and write up something coherent (and I'll help troubleshoot!).

Expected Input: Highly variant – it depends on the question you want to tackle! It'd be great if you have ideas for questions you want to analyze!

Expected Output: At a minimum, some semi-formal documentation of your analysis.

Difficulty: 🌶🌶

Prerequisites: Statistical Modelling, the class

Steve's Projects

0.47. **Introduction to Forcing.** (Steve + Susan)

Description: Forcing is a proof method in set theory for building models of the ZFC axioms with prescribed properties — for example, models in which the Continuum Hypothesis is false (or true). Forcing is even used (in combination with other techniques) to produce models in which the axiom of choice fails. This project will be devoted to learning what forcing is, and some basic arguments involving the technique; or, for campers with background in forcing, the more advanced technique of *product forcing*.

Structure: Guided reading, basically, but very difficult stuff

Expected Input: Lots of reading and working exercises (we have notes), probably multiple longish meetings per week

Expected Output: A poster is reasonable but not necessary

Difficulty: 🌶🌶🌶🌶

Prerequisites: Solid grasp of basic set theory

Susan's Projects

0.48. **Directed Graphs and Noncommutative Rings.** (Susan)

Description: How much do you know about noncommutative rings? Maybe you've played with the quaternions, or matrix rings, but there are wilder rings in this world. Consider the **free algebra on n generators**, denoted $k\langle x_1, x_2, \ldots, x_n\rangle$. This is the set of polynomials in n non-commuting variables—that is, the monomials $x_i^2 x_j$, $x_i x_j x_i$, and $x_j x_i^2$ are all distinct. This is one of my favorite rings, because by imposing additional relations on the structure, we can create new algebras with their own interesting behaviors. For example, if we start with $k\langle x, y, z\rangle$, the free algebra on three generators, we can impose the relation $xy = yx$ to obtain a ring in which x and y commute with each other, but not with z. If we impose the relations

$$x^2 = y^2 = z^2 = xyz = -1,$$

then we obtain the quaternions. We can use these sorts of relations to build rings that satisfy all sorts of cool properties. But the real fun starts when we introduce combinatorial objects like graphs.

If we have a directed graph $G = (V, E)$, we can consider the free algebra $k\langle E\rangle$, generated by the set of directed edges. Given a path (a, b, c, d) in this algebra, we can use the noncommutative multiplication $abcd$ to "talk about" the path in an algebraic way. Then we can introduce relations that roughly say that two paths are "equal to" each other if they start and end in the same place.

In this project we will be attacking an open problem in my field of research, involving a simple algebra associated to a class of objects called layered graphs. This will definitely involve some fun algebra and combinatorics. And since it's me, it's a good bet that posets will be involved as well.

Structure: This class will be a significant time commitment. We will be meeting once a day during Week 3, to learn enough background to get ourselves into uncharted territory. We will be meeting once a day during Week 4 to run a research group.

Expected Input: Time, creativity, patience, and cooperation.

Expected Output: I think we could make some genuine progress on this problem—I would love to turn the results into a paper this coming year.

Difficulty: 

Prerequisites: Basic linear algebra. Ring theory would also be a plus.

The Staff's Projects

0.49. **Teaching a Week 5 class.** (The Staff)

Description: Do you have an idea for a class you'd like to teach? Find a staff member and describe your idea — be sure to have an idea of the outline and the punchline of your potential course. Feel free to approach any staff person who you think is a good mathematical fit. Particular staff members who may be interested in supervising such a project are: Alfonso, Marisa, Mark, and Pesto.

Structure: You will spend most of your time discussing with your advisor and planning your class, and you'll teach it in week 5!

Expected Input: In the first few weeks you'll have to spend a good amount of time figuring out exactly what you want to include in your 50 minutes of time for your class – that's shorter than you might expect! You'll also spend some time outlining the class and preparing it as well as giving two practice talks before the real class. You will give a practice talk for your staff sponsor some time in Week 3, and after receiving their feedback you'll give a second practice talk some time in Week 4. Assuming that goes well, you'll be able to teach in Week 5.

Expected Output: A spot in the Week 5 class schedule!

Difficulty:

Prerequisites: Depends on what you want to teach.

Waley and Wanqi (and Mira)'s Projects

0.50. **Improving Math Education in High School.** (Waley and Wanqi (and Mira))

Description: Have you ever wanted to make the high school math curriculum more interesting and engaging, but never had the opportunity to do so? Do you want to do a summer leadership project in mathematics education? Then join Mathcamps first ever student-run project — by Mathcampers, for Mathcampers!

Last year, six Mathcamp alumni founded one of North Americas largest student-led math organizations. Two of them, Waley and Wanqi, are at Mathcamp this year and will be the leaders of this project. Although Mira is the official staff sponsor, she will mostly just provide logistical support and watch from the sidelines.

Many people dislike math. Why? Because they never learned the essence of it. In this project, you will be collaborating with Waley and Wanqi in designing activities to make the high school math learning experience more engaging. Instead of presenting math as repetitive — categorizing problems into patterns and blindly applying formulas without understanding why — you will help make math fun and understandable for many students your age.

Structure: Staff interaction on this project will be minimal. Instead, you will maintain frequent contact with the projects student coordinators, Waley and Wanqi, to update them on your progress and discuss future plans and activities. We also plan to have regular project meetings, where everyone involved with the project will gather to touch base, brainstorm, and discuss each other's ideas.

Beyond that, though, you have almost full control over how you wish to contribute to this project! You can work alone or collaborate with other students. And the more participants, the better!

Expected Input: You will be collaborating with other campers and Waley and Wanqi to design class sessions (containing activities or other interactive experiences) to creatively teach math concepts from the grade 8–12 mathematics curriculum, without breaking Rule 2. Details will be provided via email and meetings and are flexible. We are looking for people who are willing to work at least three hours a week, from Week 3 onward.

Expected Output: Lesson plans and activities that might eventually get used by teachers all over the continent! We already have a few teachers who are interested, and are working to find more.

Difficulty:

Prerequisites: Some experience tutoring or helping others learn math would be helpful.