PROJECT DESCRIPTIONS — MATHCAMP 2023

Contents

Allison's Projects

Flow Free (Allison)

Description: Have you ever played the mobile game Flow Free? I have. In fact, I currently have a 590-day streak of doing the daily puzzles. Flow Free presents you with a grid in which some squares are filled with coloured dots. You must connect the dots of the same colour in such way that no paths intersect and all space is used. Let's solve some puzzles! Solve some puzzles of various variants! Come up with our own puzzles! I'm especially interested in figuring out how to build a uniquely solvable puzzle, but am happy to take this project in other directions.

Structure: An introductory meeting to talk through what we want to do and decide on a structure, and probably weekly meetings after that.

Expected Input: Puzzling, in the quantities that will make you happy.

Expected Output: A very good intuition for solving Flow Free puzzles, a new hobby, and some puzzles of our own.

Difficulty: 🌶–🌶🌶

Prerequisites: None

Allison and Gloria's Projects

Make something! (Allison and Gloria)

Description: This is not a project in the same sense as the others: in particular, you can participate even if you are participating in another project! We have a wonderful crafts lounge in Valcour with many materials for making beautiful and interesting mathematical objects. You can knit or crochet things as simple as hyperbolic planes or as time-consuming as a Klein bottle hat, or Mobius strips, or many other objects. You could design and make a quilt. Even collaborate on a larger project in a group! Ask Allison or Gloria for more ideas, look at the crafts lounge, and/or look here: `https://tinyurl.com/mathyyarnideas`. No need to know how to knit, crochet, tat, or quilt beforehand; we will help you learn!

Structure: Individual or group projects are fine. If you are interested, stop by the crafts lounge, or ask Allison or Gloria.

Expected Input: Between one hour and lots

Expected Output: Fun objects

Difficulty: 🌶

Prerequisites: none

Ania and Mia's Projects

Plan a math circle session (Ania and Mia)

Description: Would you like to learn more about how to communicate math? Are you thinking about maybe running a math circle or doing other teaching outside of camp? Are you excited about teaching, but don't necessarily want to teach a whole Mathcamp class? Then this project is for you!

In this project, you'll design and plan a math circle session on a topic you are excited about. The session can be a lecture, a problem set, or even an IBL worksheet—it's up to you! You'll spend time thinking about how to convey mathematical ideas, engage an audience of students, refine a class plan and more. If you want, you can then present your class during a schedule board event or during the Project Fair (but it's completely optional).

Structure: Meet 1–2 times a week to talk about the progress, work between the meetings with your group.

Expected Input: At least one meeting a week and 2+ hours of individual work in between the meetings (and can be much more if you're excited!)

Expected Output: A class plan / problem set / IBL worksheet, possibly a short informal session taught by you.

Difficulty: 🌶–🌶🌶

Prerequisites: None

Arya's Projects

Brethren of travelling pants (Arya)

Description: Our object of focus is the sphere with 5 holes—topological overalls :) A "pants decomposition" for this surface is any pair of disjoint loops that don't bound a cylinder (why?). The "pants graph" is a graph constructed using these pants decompositions—the vertices correspond to pants decompositions, and edges correspond to certain "elementary moves".

The goal of this project is to study this "pants graph" in this setting. In particular, given two pants decompositions, we would like to find paths in this graph from one to the other. This project would involve drawing many pictures and testing examples, and would be a great introduction into modern geometric topology.

Structure: Meetings in TAU (at least once a week) and some time spent individually doing computations.

Expected Input: Drawing many pictures and testing examples, occasionally reading and discussing research papers.

Expected Output: Hands-on introduction to geometry and topology of surfaces, possibly leading to original research.

Difficulty: 🌶🌶🌶🌶

Prerequisites: Some familiarity with surfaces (or general topology) would be helpful. Talk with me!

Hyperbolic hyperbolic geometry (Arya)

Description: Hyperbolic geometry is the coolest math anyone can ever imagine. The exuberant pictures of thin triangles, wonky circle inversion being an isometry, and angles determining area will literally blow your mind to bits. What's more, almost every surface, graph or group is hyperbolic—hyperbolic geometry is no cap the only kind of geometry that actually matters.

In this project, you shall learn the entirety of millennia of geometric research in three short weeks. After the three weeks, you shall reach philosophical and mathematical enlightenment, of a kind never seen before.

Structure: Read notes, present in weekly meetings, possibly with homework exercises.

Expected Input: About an hour or two of self-study per week, with weekly meetings during TAU.

Expected Output: Some understanding of hyperbolic geometry, hopefully with understanding the geometrisation of surfaces.

Difficulty: 🌶–🌶🌶

Prerequisites: Some familiarity with complex numbers and surfaces. Some familiarity with groups would also be useful.

Ben's Projects

Classical cryptanalysis (Ben)

Description: In 2015, a paper by Alan Turing (1912–1954) appeared on the arXiv, an online repository of math papers.[1]

The paper discusses how to use probability to attack the Vigenère cipher, a commonly-used encryption method that is broadly similar to the Enigma cipher.[2] In this project, we'll try to implement Turing's methods to break this cipher, and possibly other ciphers, either by computer or by hand.

Structure: Reading Turing's notes, thinking about what you want to implement, and meeting to discuss these roughly weekly.

Expected Input: A few hours of work a week, either reading or implementing

Expected Output: More experience with probability and cryptanalysis

Difficulty: 🌶🌶

Prerequisites: None; probability not required ahead of time. If you want to do this on the computer, prior experience coding would be useful.

Throwing stuff and catching it (Ben)

Description: Want to learn a bit about juggling, and how to find new and fancy (or incredibly silly) ways to juggle? It turns out that there's actually a little bit of math in juggling!

[1] Due to more of his notes becoming available, not necromancy. In case you were concerned.

[2] The Vigenère cipher is way simpler, though

This might be the project for you if you want to investigate some of that math, or just learn how to juggle, or anything in between!

Structure: Meeting weekly to discuss math of juggling (I'd introduce site-swap notation at the first such meeting, set a few "standard" problems, and see where things go from there)

Hopefully also some meetings where we learn to juggle.

Expected Input: A bit of time scribbling about some juggling math. Maybe also a bit of time spent juggling. Hopefully a decent amount of time spent juggling!

Expected Output: Either a poster describing some site-swap facts or—possibly—some campers who can juggle some of the less standard patterns as a demonstration!

Difficulty: 🌶–🌶🌶

Prerequisites: None—neither preexisting ability to juggle, nor willingness to juggle are strictly required here. I suspect this project will be more fun if you're at least willing to learn to juggle, however.

Ben and Steve's Projects

Abstract functional analysis reading project (Ben and Steve)

Description: This project will build more-or-less directly off of the week 3 class "Calculus of variations." There's a recent book (*The convenient setting of global analysis* by Kriegl and Michor) which presents a very abstract and general approach to calculus of variations-type problems. It looks really interesting, and I'd love to work through (part of) it with some campers.

BE WARNED: We will be learning alongside you all, so... should be fun!

Structure: We'll work through a book

Expected Input: A couple hours a week

Expected Output: Probably nothing; maybe a poster?

Difficulty: 🌶🌶🌶🌶

Prerequisites: Calculus of variations, multivariable calculus, linear algebra

Charlotte and Narmada's Projects

Crossover episode (Charlotte and Narmada)

Description: Linear Algebra meets Analysis! Narmada meets Charlotte! Hahn meets Banach!

This will be a reading project in functional analysis. If you liked linear algebra in finite dimensions, think again. Analysis makes it worse. We'll see how the Axiom of Choice saves us from losing it all.

Structure: Meeting once or twice a week to talk about readings

Expected Input: Approx 2 hours reading each week (outside of meetings)

Expected Output: Poster for the project fair

Difficulty: 🌶🌶🌶

Prerequisites: Linear algebra, some understanding of continuity

Della's Projects

Build a Turing machine in Minecraft (Della)

Description: Lots of people have built computers in Minecraft. Some of them are even structured like Turing machines, and people claim Minecraft is Turing-complete. But—as far as I know—they're all limited in that they have a fixed amount of memory, and giving them more memory requires making them bigger.

I want to fix this! It seems like it should be possible to build a flying contraption that acts as a Turing machine, and also builds more tape as needed (using blocks from a cobblestone generator). This would be a finite structure that has infinite memory, which is more powerful than something with finite memory. The goal of this project is to build such a contraption.

I have ideas for how a lot of the pieces of this might work, but not all of it, and there's a lot of design work to do. This would involve thinking about Turing machines, and designing large and intricate flying machines in Minecraft.

In the language of computational complexity theory, existing Minecraft computers prove that Minecraft is PSPACE-complete, but a true infinite-memory Turing machine would prove it's Turing-complete (or RE-complete), provided we pretend that Minecraft worlds don't arbitrarily end at 30 million blocks.

Structure: I'll meet with you share my thoughts on how this could work and tell you about Turing machines if needed. Then you'll work mostly independently, checking in with me once/week or as needed when you have progress or feel stuck.

Expected Input: As much time as you want—I don't think this can reasonably be completed in one Mathcamp, but you should be able to make significant progress with 4 hours/week.

Expected Output: Partial progress towards a Turing machine in Minecraft. This might mean solving subproblems (e.g. a flying machine that builds more tape on demand) or a high-level plan for how the pieces will fit together once they're designed in more detail.

Difficulty: 🌶🌶🌶

Prerequisites: You should be familiar with Minecraft, in particular redstone and ideally slimestone. It would help, but isn't necessary, if you've seen Turing machines before.

Read a paper about gadgets (Della)

Description: There's more to say about gadgets than fits in a one-week class! Gadgets are a very young research area, so they only place to learn about them outside of Mathcamp classes is by reading papers. If you enjoyed Inspecting Gadgets, you can do that!

There are several papers answering different questions about gadgets: which of the 60 planar embeddings of the door can make a crossover? How do we define simulations more formally than I did in Inspecting Gadgets? I'll help you pick a paper based on your interests and background, and then you'll read it!

Structure: I'll meet with you to help you pick a paper, and then you'll work through it at your own pace. We'll talk once/week or as needed to go over what you've read and answer any questions that come up.

Expected Input: 1–4 hours per week, based on your preference and which paper you choose.

Expected Output: If you'd like, a project fair poster about what you learned!

Difficulty: 🌶🌶

Prerequisites: Talk to me first if you didn't take Inspecting Gadgets.

Solve open problems about gadgets (Della)

Description: Since people haven't been thinking about gadgets for very long, There are a lot of approachable open problems. If you enjoyed Inspecting Gadgets, you can try to solve some of them!

Possible directions to think about include

- Is there a gadget which can simulate every balanced gadget? What about other simulability classes?
- In the other direction, can we characterize the class of gadgets the 1-toggle can simulate? The dicrumbler?
- Can you build interesting gadgets in your favorite video game?

If there's something you're interested in that isn't one of those, that's fine too! We'll meet to talk about what you want to think about, and I'll tell you about any relevant known results.

Structure: After I help get you started, you'll work mostly independently, and meet with me once/week or as needed to talk about progress and answer questions.

Expected Input: Flexible, depending on interest.

Expected Output: If you'd like, a project fair poster about what you discovered!

Difficulty: 🌶🌶🌶

Prerequisites: Talk to me first if you didn't take Inspecting Gadgets.

Eric's Projects

Combinatorial music (Eric)

Description: The goal of this project is to learn various pieces of modern classical music that fit into the description of "combinatorial music". The most famous of these is Steve Reich's "Clapping Music" which you should Google if you want a sense of the flavour of these events. Eric has a few other pieces of music queued up, and we can devote some time to researching what is out there and finding other pieces.

Structure: Meeting at TAU or in the evening a couple of times each week to practice.

Expected Input: Meet for at most 2 hours a week to discover, transcribe, and practice various pieces.

Expected Output: Enjoyment! Perhaps also a performance at the talent show.

Difficulty: 🌶

Prerequisites: Familiarity with music notation. You don't need to be able to sing or play an instrument.

Practice your algorithmic thinking with advent of code (Eric)

Description: Advent of code (`https://adventofcode.com/`) is a yearly series of programming challenges that can be solved in any language. There's 25 puzzles, each with two parts; the second part is a twist on or extension of the first. The first few puzzles are simple, and can be solved in just a few lines of code. Later on the solutions can become quite involved and require very careful thinking through the algorithm used in order to get a solution that runs in a reasonable timeframe.

Structure: We'll have regular check-ins about once a week, and you'll do most of the working on your own.

Expected Input: Campers participating in this project are expected to work on the problems in a self-guided way, with Eric available for consultation and regular check-ins.

Expected Output: Practice your algorithmic thinking skills!

Difficulty: 🌶🌶🌶–🌶🌶🌶🌶

Prerequisites: You should have written a computer program before. Eric is happy to supervise you working in basically any programming language, especially Python or Haskell.

Eric and Narmada's Projects

Math Camp Student Parsely (Eric and Narmada)

Description: COMPUTER: You are holding the BLURBS for the project selection fair in your hands. What would you like to do?

> READ BLURBS: You flip through to the first blurb that catches your eye. "Welcome to our BLURB! Have you ever felt slighted by the lack of Mathcamp-themed choose-your-own-adventure games? Slight no longer! Speak to ERIC and NARMADA at the project selection fair for more details."

> READ ANOTHER BLURB: You try to read some of the other blurbs but you can't stop staring at this one.
> SPEAK TO ERIC AND NARMADA: You're not at the project selection fair yet.
> GO TO THE PROJECT SELECTION FAIR: See you there!

Structure: The campers will produce a Mathcamp-themed Parsely-style choose-your-own-adventure game

Expected Input: Meetings twice a week

Expected Output: Run a scheduleboard event to play the Mathcamp game

Difficulty: 🌶

Prerequisites: none

Eric and Tim!'s Projects

Learn change ringing (Eric and Tim!)

Description: Change ringing is the musical tradition of performing symmetric groups on bells. You might have seen Eric and Tim! Running events about it or heard about it in Eric's week 1 group theory class. This project is your chance to learn it for yourself! It is a wonderful combination of music, mathematics, concentration, and meditation. We'll teach you some basic ringing patterns and how to perform them in practice. Optionally, we can talk about the mathematics behind the patterns we choose to ring.

Structure: We'll be practicing ringing using a set of handbells.

Expected Input: Attend practices several times a week. This project requires a regular commitment to practicing, but is not spicy otherwise.

Expected Output: Poster at project fair, performance at the talent show.

Difficulty: 🌶🌶

Prerequisites: None!

Ian's Projects

Mathcamp Sudoku Project (MCSP) (Ian)

Description: How skilled are you at Sudoku—do you have the techniques that can solve all the solvable Sudokus? We plan to gather a number of times during the week to work on the New York Times daily Sudoku, easy to hard, and share different techniques we have in solving the Sudoku puzzles.

Structure: We gather in a room or a place with a projector. Having embedded a huge Sudoku board on the whiteboard, we will collaborate to work on the daily Sudoku puzzles. We will split the campers into three groups based on difficulty, so that campers in the corresponding group have priority in speaking. Ian will mediate the conversation so that only one person is speaking at a time instead of having a huge brawl among the campers.

Expected Input: Contribution to solving Sudoku puzzles.

Expected Output: Undecided, but maybe a presentation on different techniques we learned throughout this project?

Difficulty: 🌶🌶–🌶🌶🌶

Prerequisites: Knowledge on basic rules of Sudoku

Kayla's Projects

Join the Cluster Clan (Kayla)

Description: We will be exploring the land of cluster algebras in this project! These are certain types of algebraic structures that are governed by combinatorial rules. A cluster structure takes a piece of initial data—e.g. a triangulated polygon, a directed graph, or a topological surface—a set of variables, and a mutation rule that prescribes a way to transform this data to create something new. The set of all things you can generate via this mutation rule itself has even more structure! Enter cluster algebras. This abstract phenomenon of initial seeds and mutation leads to a beautiful intersection of algebra, combinatorics, topology and geometry.

Structure: We will read through some old lecture notes of mine and then tackle a paper my Gregg Musiker and Lauren Williams about matrix formulae for cluster algebras.

Expected Input: 1–2 hours outside of meetings reading and coming up with examples to talk about and present.

Expected Output: Familiarity with the definition of cluster algebras + a taste of algebraic combinatorics

Difficulty: 🌶🌶🌶

Prerequisites: Linear algebra: specifically matrix multiplication and trace; familiarity with some group or ring theory would be nice!

Join the Cluster Clan *extreme addition* (Kayla)

Description: In this project, we will be studying the world of cluster algebras through an intense algebraic lens. One perspective of cluster theory is through the world of quiver representations. A quiver representation (take my and Raj's week 4 class) is a linear structure that one can put on a directed graph. Studying these representations is pictorial and pretty concrete, which makes them a fun object of study. But are they useful. . . ? It turns out they are extremely useful! Every representation of a certain types of algebraic structures can be realized as a quiver representation! This got people thinking. . . what else can quiver representations help us with? When bootstrapping the idea of quiver representations to a categorical framework, we can create an algebraic point of view for cluster theory that has proven extremely useful in the field. Hold on campers, this one will be *extremely* fun!

Structure: Reading at home with discussion during meetings.

Expected Input: 1–2 hours of focused reading before each meeting. Meeting twice a week!

Expected Output: Seeing one example of the cluster category!!

Difficulty: 🌶🌶🌶🌶–🌶🌶🌶🌶🌶🌶

Prerequisites: linear algebra and group theory

Kevin's Projects

Learn German (to read a paper!) (Kevin)

Description: I don't know any German. In this project, we get to struggle together to learn some German and maybe even some math by reading (and possibly translating, if we have the energy/time) a classic paper. Options include papers by Riemann, Noether, and Hurwitz from the 1800's and early 1900's (but I'm open to suggestions!). Plenty of Google Translate will be used!

Structure: Meet with Kevin at least twice a week at TAU or in the evening

Expected Input: 2+ hours per week

Expected Output: A translation of (part of) a classic paper if we have the energy/time; otherwise just some fun math

Difficulty: 🌶🌶🌶–🌶🌶🌶🌶

Prerequisites: Some real/complex analysis or group/ring theory (for the papers I have in mind), but it definitely depends on the paper

Kevin and Raj's Projects

Quantum groups and crystal bases (Kevin and Raj)

Description: In this project, we will learn about quantum groups and crystal bases, following *Introduction to Quantum Groups and Crystal Bases* by Hong and Kang (or whatever other resources we find). Quantum groups are ubiquitous objects in mathematical areas ranging from mathematical physics to representation theory, knot theory, and combinatorics. Crystal bases provide a powerful combinatorial tool to help understand representations of quantum groups and Lie algebras. We hope that this project will serve as an introduction to some cool parts of representation theory and combinatorics.

Structure: Kevin and Raj will meet with you at least twice a week to talk through the project. We want the pace to be such that everybody is on the same page.

Expected Input: Up to you.

Expected Output: An expository write-up of the material.

Difficulty: 🌶🌶🌶🌶

Prerequisites: Required: Group theory, ring theory, linear algebra (including multilinear algebra: tensor products, duals, symmetric powers, etc.). Helpful: Some familiarity with Lie algebras and root systems.

What is Schubert calculus? (Kevin and Raj)

Description: In this project, we will learn about modern solutions to classical intersection problems, such as "How many lines intersect four given lines in space?" Such intersection problems are notoriously difficult to solve without using tools from algebraic topology and algebraic combinatorics. The field we will be studying is known as Schubert calculus.

We will begin the project by analyzing the Grassmannian of k planes in space. In particular, we will show that the Grassmannian is the zero set of a system of polynomial equations. We will then look at how the Grassmannian breaks up into pieces called Schubert cells. Understanding the intersection theory of Schubert cells is the key to understanding the classical intersection problems mentioned at the beginning.

We will be following several sources, such as Fulton's *Young Tableaux*. No matter how far we get, we hope that the project will serve as a nice introduction to both algebraic topology and algebraic combinatorics.

Structure: Kevin and Raj will meet with you at least twice a week to talk through the project.

Expected Input: Up to you.

Expected Output: An expository write-up of the material.

Difficulty: 🌶🌶🌶🌶

Prerequisites: Linear algebra

Krishan's Projects

Theorem proving in Lean (Krishan)

Description: Proofs are an essential component of mathematics, but how can we know if a proof is correct? One way to do this is to explain your proof to a computer and ask it to check that each step follows from the last. Luckily there is software that can do just that. We'll start the project by working through the Natural Number Game developed by Kevin Buzzard and Mohammad Pedramfar. The game is all about proving basic facts about the natural numbers like $0 \cdot n = n$ in a way that a computer can understand. Once we get through the game, we'll begin formalizing more complex mathematical statements and proofs using the Lean proof assistant.

Structure: In small groups or pairs the students will work their way through the Natural Number Game for the first week or so. Then I'll help the student install Lean and together we will work on exercises from the textbook, Mathematics in Lean.

Expected Input: About 3 hours per week

Expected Output: An understanding what formalization in mathematics is all about and the ability to keep learning more about the subject. A motivated student could keep working on this after Mathcamp and could formalize a proof that has never been formalized before.

Difficulty: 🌶🌶

Prerequisites: Some experience coding (+1 chili otherwise)

Mia's Projects

Create your own adventure, literally... or should I say, pictorially! (Mia)

Description: In the past two decades, graphic novels have enjoyed a surge in popularity and have been increasingly recognized as a powerful form of literature. Moreover, they've also proven to be a fascinating model for mathematical story-telling and exposition. In this project, you'll explore some recent mathematical graphic novels (and comics!) and use those as inspiration to make one of your own.

Structure: We'll spend the first week reading and talking about examples of mathematical graphic novels. From there you'll work on drafting, sketching, and producing one of your own!

Expected Input: Expect to spend a couple of hours reading and discussing graphic novels during the first week. The following weeks will be spent producing your own graphic novella or comic series, so the time commitment will vary; it could range from 4 hours to significantly more!

Expected Output: A mathematical graphic novella or comic series.

Difficulty: 🌶–🌶🌶🌶

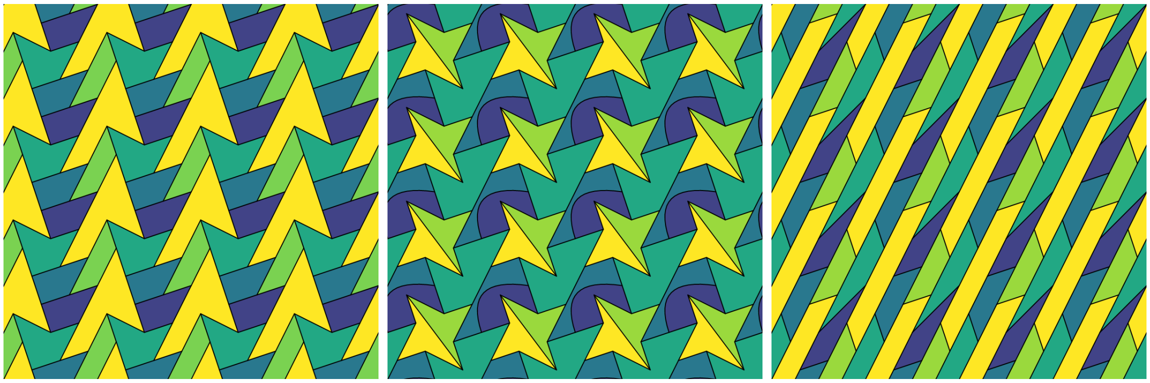
Prerequisites: None

Misha's Projects

Drawing toroidal maps (Misha)

Description: If you took Tim!'s class on graph theory, you might have encountered the Petersen graph. If you didn't take Tim!'s class on graph theory, you might also have encountered the Petersen graph; it is very famous.

The Petersen graph is not planar, but it can be drawn without crossings in the torus. We can think of a torus as a square with opposite edges glued together, or we can put together many identical squares into a tiling of the plane. Here are three tilings that correspond to toroidal embeddings of the Petersen graph:



(The first of these was found by Mathcamp alum Drake Thomas.)

In this project, you will draw more toroidal embeddings of notable graphs that have genus 1 (in other words, they can be embedded in the torus, but not the plane). It is satisfying to find embeddings that are particularly simple, regular, or symmetric, but it is slightly up to you to figure out what that means.

Structure: I think it's more fun if you explore independently, but I can suggest some particular goals it might be worthwhile to try for. Separately from the mathematical and creative steps, I can also show you how I made these drawings and/or help you create your own.

Expected Input: At least 1 hour/week, possibly more.

Expected Output: Some nice drawings of nonplanar graphs.

Difficulty: 🌶🌶

Prerequisites: Some graph theory might be nice for motivation and intuition, but I can catch you up.

Master countless short papers (Misha)

Description: There are often Mathcamp projects that are about reading and understanding research papers. But why stop at one?

Of course, Mathcamp is short, and if you want to read many papers, you have to be strategic. I have picked out several particularly short papers: one page long (sometimes front and back, and in a couple of cases only if we skip the references). You will try to get through as many of them as possible.

Of course, we need to define what counts as having read a paper. Let's say that you should explain the results in the paper to me, and if necessary explain why they're interesting, using no references except possibly notes you wrote for yourself.

Structure: I print paper, I give you paper, you explain paper, I print next paper.

Expected Input: As much time as you're willing to put in. (There are more interesting short papers out there than you can read in the remaining hours of Mathcamp.)

Expected Output: You will understand some neat short proofs. If you like, you can prepare a presentation about some of them for the project fair in week 5.

Difficulty: 🌶🌶🌶🌶

Prerequisites: For the papers I have selected so far, some knowledge of calculus, graph theory, group theory, and number theory is helpful (in no particular order), but I can find more papers, and can try to aim for prerequisites you have.

Soma Cube origami (Misha)

Description: The Soma cube is a puzzle invented by Piet Hein and popularized by Martin Gardner. Seven 3-dimensional pieces (each made up of three or four cubes) can be put together into a $3 \times 3 \times 3$ larger cube (as well as other shapes).

In this project, you will make the seven Soma pieces out of origami and figure out how to put them together into a cube. If you're interested, you can continue to solve other puzzles with these pieces, compose your own puzzles, or maybe explore what can be done with puzzle pieces of other shapes.

Structure: You can work alone or in groups; if you work in a group, you can finish the puzzle pieces faster, but you might eventually want to each have your own set anyway.

If there are multiple participants, one fun way to continue the project is to compose puzzles for each other.

Expected Input: I bet 1–2 hours of concentrated work is enough to make the Soma cube, but you can work at a more laid-back pace if you like.

Once you finish the Soma cube, the pace of the project from there is up to you!

Expected Output: One or more Soma cubes; some puzzles or solutions to them.

Difficulty: 🌶

Prerequisites: None; it will help to have some origami experience, but the origami side of this project is not technically difficult.

Narmada's Projects

Become ambidextrous (Narmada)

Description: Title says it all.

Structure: We will meet as needed to practice using our non-dominant hand

Expected Input: One non-dominant hand

Expected Output: Zero non-dominant hands

Difficulty: 🌶🌶🌶🌶

Prerequisites: Don't be ambidextrous

Neeraja's Projects

An algorithm for the Borsuk–Ulam theorem (Neeraja)

Description: In this project we will read this short paper (https://www.ams.org/journals/proc/1979-073-01/S0002-9939-1979-0512075-9/S0002-9939-1979-0512075-9.pdf), which gives an elementary and constructive proof of the Borsuk–Ulam theorem. Depending on how ambitious we want to be, we can also aim to implement the algorithm giving the solution in a computer program. Or, if we want to be a bit more relaxed, we can just focus on understanding this three-page proof.

Structure: First two weeks of the project: reading the proof. Depending on how this goes, we either continue reading the proof in the third week or we can try to implement the algorithm in the third week.

Expected Input: 3–4 hours per week, but more if we decide to implement the program

Expected Output: Maybe a program, otherwise could be a presentation by the students

Difficulty: 🌶🌶–🌶🌶🌶

Prerequisites: Uniform convergence, a little experience with proofs. "Epsilons and Deltas" by Ben and Charlotte would be enough.

Philosophy of math (Neeraja)

Description: We'll read a text in the philosophy of math, possibly Imre Lakatos' *Proofs and Refutations* or Ludwig Wittgenstein's *Tractatus Logico-Philosophicus*, or a different text that you choose!

Structure: Reading and discussion

Expected Input: 3–4 hours per week

Expected Output: Maybe a presentation or even a short essay depending on interest

Difficulty: 🌶

Prerequisites: None

The Szemerédi–Trotter theorem in incidence geometry (Neeraja)

Description: The Szemerédi–Trotter theorem is an important and widely used result in incidence geometry. Given n lines and m points in the plane, the theorem counts the maximum number of incidences (i.e. the number of point-line pairs, where the point lies on the line) that can occur. This project will involve reading the proof of the Szemerédi–Trotter theorem. There are different proofs—one involves a cell decomposition and one uses graph theory. We can choose which one to read based on your tastes!

Structure: Reading the proof and meeting with Neeraja at least twice a week to discuss

Expected Input: 3–4 hours per week

Expected Output: I'd suggest a poster, since this theorem is really nice to illustrate.

Difficulty: 🌶🌶

Prerequisites: None

Steve's Projects

Logic reading: nonclassical propositional logics (Steve)

Description: Propositional logic is usually presented as a pretty simple thing: without diving into the "internal structure" of mathematical sentences (which we need to do to make sense e.g. of quantifiers "for all" and "there exists"), there are only a few ways we can combine/modify such sentences, namely AND ($\wedge$), OR ($\vee$), NOT ($\neg$), and the other Booleans. If we want to get really fancy we can point out the connection with *partially ordered sets* (and *Boolean algebras* in particular), but ultimately the picture is fairly limited.

This, however, is only because we've chosen to focus on **classical** propositional logic. What if we took a more nuanced view of negation (e.g. maybe "not not X" is different from "X"), or had more truth values than just TRUE and FALSE, or allowed a more complicated notion of implication than that given by the usual truth-table? What if we tried to incorporate *modalities* such as "— is necessarily true" ($\Box$) and "— is possibly true" ($\Diamond$)? And so on.

It turns out that propositional logic**s** themselves are quite intricate mathematical objects once we go beyond the merely classical. In this reading course we'll learn the basics of one particular approach to nonclassical logic, namely the abstract algebraic logic of Blok, Pigozzi, and others. We'll also treat other cool things as they come up.

[NOTE: if you're interested in doing a logic reading course on a different topic, ask me; I might be amenable to that!]

Structure: I'll provide readings, exercises, and help.

Expected Input: A couple hours a week.

Expected Output: Probably nothing, but if you want to make a poster I could help with that.

Difficulty: 🌶🌶🌶 – 🌶🌶🌶🌶

Prerequisites: Solid understanding of Boolean logic; e.g. you should know the truth table of the material implication.

Multiplayer combinatorial games (Steve)

Description: Combinatorial game theory is highly developed in the case of 2-player games. 3-player games, however, are much less studied; I'm only aware of a couple papers on the topic, and there are a number of questions which I think are approachable but seem unanswered. And of course things only get more mysterious for n-player games as $n \to \infty$.

In this project we'll look into how the various ideas of basic combinatorial game theory do (or don't) extend to games with three-or-more (but only finitely many!) players. Along the way, we might want to revisit the basic definitions of 2-player combinatorial game theory and modify them to suit emerging intuitions.

Structure: We'll start with a quick high-level summary of classical 2-player combinatorial game theory and some motivating examples/questions about games with more players; from there, it's up to the student, although I have several suggestions. The main goal is to develop a set of definitions and "test questions" which make sense and are interesting; actually resolving some of those questions would be nice, but is a secondary focus.

Expected Input: A couple hours a week.

Expected Output: Could be none, could be a poster or a short write-up

Difficulty: 🌶️🌶️🌶️–🌶️🌶️🌶️🌶️

Prerequisites: Understanding of basic combinatorial game theory, e.g. computing nimbers and/or red/blue Hackenbush position values.

Tanya's Projects

Mathematical literature or literary mathematics: selections from "Labyrinths" by Jorge Luis Borges (Tanya)

Description: The acclaimed Argentinian writer Jorge Luis Borges weaved mathematical concepts into many of his short stories from the famed collection titled "Labyrinths" in English. For instance, "The Library of Babel" explores the idea of what would happen if one generated random strings of texts ad infinitum. Would you eventually produce a famous work of literature, such as Hamlet? (The answer is "yes" by Borel–Cantelli :)) In this project, we will read a selection of stories from the Labyrinths collection, understand which mathematical concepts arise, how Borges presents them and, conversely, how mathematics is being used as a literary device. We will also attempt to write our own short stories based on concepts learned at Mathcamp! My hope is that this experiment will allow us all to look at mathematics from a novel more poetic perspective.

Structure: We will read and discuss a sample of short stories in the first two weeks of the project, then work on writing our own stories in the spirit of Borges.

Expected Input: About 1–2 hours of reading and 1 hour of discussion in the first two weeks. The final two weeks may require greater time commitment, but perhaps less structured. I would anticipate writing a well thought-out 1000 word short story may take 5–6 hours of your time over the course of the two weeks.

Expected Output: A creative short story exploring a mathematical concept in a poetic way that each student likes!

Difficulty: 🌶️

Prerequisites: None

Mathematics and its impact on society: "Weapons of Math Destruction" by Cathy O'Neil (Tanya)

Description: We are all here at Mathcamp because we love how conceptually beautiful mathematics is. However, math is not merely a collection of puzzles we solve "for fun"—it is also incredibly important for many areas of science and engineering, and, in particular, data science, a field that is currently drawing a lot of resources and attention and, furthermore, permeates our daily digital (and sometimes analog) existence. Cathy O'Neil, a former number theorist and Wall Street quant turned ethical AI advocate, wrote a fascinating book on how dangerous these algorithms can be and how they co-opt our beloved discipline along the way called "Weapons of Math Destruction". I have been meaning to sit down and read this book for a while and I would love to do that together with other students at Mathcamp and any interested staff! I think understanding the risks/limitations of our society's current obsession with tech and data is incredibly important for all of us conducting research/teaching/studying dangerously close to these areas.

Structure: We would read a few chapters of the book every week (about 1–2 hours) and then meet for a discussion (about 1 hour).

Expected Input: We would read a few chapters of the book every week (about 1–2 hours) and then meet for a discussion (about 1 hour).

Expected Output: We will read Cathy O'Neil's book and maybe look for more recent examples of what she calls "Weapons of Math Destruction"—since it was published 7 years ago I expect that we can come up with our own case studies she had not yet witnessed while writing this (this is all TBD based on enthusiasm, time etc).

Difficulty:

Prerequisites: Some familiarity with algorithms/machine learning may be helpful, but really none since the book is aimed at the general public and not a technical audience.

Tim!'s Projects

Evasiveness (and the real story of Qualifying Quiz 2022 problem 6) (Tim!)

Description: This is a reading project on *evasiveness*, a subtopic of complexity theory in theoretical computer science.

One way to measure the complexity of a problem (like "Does this graph have a Hamiltonian cycle?") is by its time complexity—roughly, how long it takes a computer to solve it. Another important way to measure complexity is query complexity—roughly, how many questions you need to ask about the graph to answer the problem. The graph properties with maximum query complexity are called *evasive*. A conjecture about the complexity of graph properties has eluded mathematicians for over forty years: the conjecture is that a huge class of graph properties—specifically, all those that are nontrivial and monotone—are evasive.

You'll trace the story of this conjecture by reading papers. The story starts in 1973, and from there it branches out—you can choose which way to go. There's a surprising appearance of topology in 1984, that has persisted through much of the more recent work. There are interesting results from within the past decade. Along the way, you'll see scorpion graphs, clever counting, collapsible simplicial complexes, transitive permutation groups, and hypergraph properties.

Problem 6 from last year's Mathcamp Qualifying Quiz (`https://mathcamp.org/2022/quiz/`) was secretly an important theorem in this area. In that problem, you were asked to locate the Mastermind of a dangerous cult by investigating the connections between the cults members. If you solved that problem, you essentially disproved an early version of the Evasiveness Conjecture. As part of this project, you'll see how that theorem fits into the bigger story.

Structure: You'll read some of the papers that moved this field forward, and we'll meet to talk about what you've read once or twice a week.

Expected Input: You will read papers and work through some of the math in those papers yourself. There's essentially no upper bound on how long you can spend reading these (there are a lot of interesting papers), but to get something meaningful out of it, you should plan to spend at least 10 hours in total.

Expected Output: Knowledge about evasiveness, and perhaps some ideas on how to explore this topic yourself.

Difficulty: 🌶🌶🌶 – 🌶🌶🌶🌶

Prerequisites: Group theory, graph theory.

Make a math video (Tim!)

Description: Communicating math in a short time to an online audience is an interesting challenge. To make a video that people will want to watch and will get something out of, you need to really understand what's compelling about your chosen topic topic, you need to be able to distill it down to its core idea. Those are both skills that are very valuable to have as a mathematician.

In this project, you will make a 3–5 minute video on a math topic of your choosing. You will choose the topic, find the compelling ideas, write the script, film it, and edit it together.

Structure: You'll can with a group or alone. We will meet about twice a week to discuss your progress and next steps, and you will meet with your group to work in between those meetings.

Expected Input: 10–20 hours

Expected Output: A 3–5 minute math video!

Difficulty: 🌶 – 🌶🌶🌶🌶

Prerequisites: None.

Mathematical engineering with Arduino (Tim!)

Description: An Arduino board is an inexpensive microcontroller designed for makers and artists. It's really good at **interacting with the real world**. It can take inputs—like light sensors, temperature sensors, joysticks, ultrasonic distance sensors, capacitive touch sensors—and turn them into outputs, like controlling a motor, making a sound, or lighting up 300 individually addressable LEDs embedded in a blue tape spider web.[3] It's designed for people who are not programmers (though if you have programming experience, that further expands what you can do with it), so this project has **no programming experience required**.

The goal of this project is to bring a mathematical concept (or non-mathematical, if you prefer) into the real world—make a game where you can move the pieces, a sculpture you can interact with, etc. You do *not* need to have a specific idea in mind to join this project: you will follow a **three-step engineering design process** to generate your final project plan.

For inspiration, here are a few of my ideas (and partial ideas). You are welcome to choose one of the ideas, but you will also get to go through the engineering design process to generate your own.

- **Guess-my-object with a lie.** This is a human-vs-computer game. There are sixteen objects, and the human player secretly chooses one of them. The game goes through seven rounds; in each round, the computer lights up some of the objects, and the player indicates whether their object is lit up by touching a capacitive touch sensor hidden in the board. The player is allowed to lie in one of the rounds (or always tell the truth if they prefer), but the computer is able to figure out the lie and identify the object.
- **Visualizing groups of symmetries.** There is a board on which you a polygon made of some solid material. The polygon has magnets embedded in it, and the board has magnet sensors in it that can detect the position of the polygon. When the user rotates or flips the polygon, that transformation is an element of a dihedral group, and that element can be written out on a two-line LCD display. This could also be done with a 3D polyhedron instead of a polygon. It can be turned into a game in various ways—for instance, the

[3] In case you haven't seen it, the blue tape spider web in the main lounge can light up and spell letters! That's all running on an Arduino board, with one other little microcontroller attached that's only needed to be able to control it wirelessly.

computer could display a group element or a product of group elements, and the user would try to perform that operation on the polygon.

- **Snake on a hyperbolic plane.** Make a hyperbolic plane (or another favorite space) and put lights on it. Then play the classic video game snake on it: The player uses a joystick to control a "snake" (a line of lit-up lights) that goes around eating pellets (other lit-up lights), growing longer as it does so, tries to prevent the snake from running into itself.
- **Nim-playing robot.** Make a robot that plays nim (or another favorite game) by sensing piece position and physically moving them with motors.
- **Visualization on a globe.** Place lights on a sphere, compute the 3D positions of the lights, and then create a visualization. There are lots of possibilities—it could be something mathematical, or something about the earth, like displaying the current weather around the world. You could control it with buttons or with position sensing in space, such as with ultrasonic distance sensors.
- **Something else.** There are many possibilities! The above are some ideas, but ideally you go through the design process to come up with an idea you are excited about.

Most of the above ideas involve some sort of math—you can bring in math you already know or learned at Mathcamp, or you can learn new math as part of the project.

This is a great project to try out Arduinos or programming for the first time (or the n^{th} time), to build something creative and mathematical, to design something with others, and to learn a bit of new math!

Structure: You will work through an engineering design process to generate ideas for something to build, then work with a small group to build one of those ideas. If you pick an idea involving math you don't know yet, you will spend some time learning it. You'll spend quite a bit of time on the physical construction. Programming can be a small part of the project or a large one, depending on the backgrounds of the people in your group, and what you are building.

Expected Input: 15–20 hours of work, working with your group on learning math, building, and programming, over 2.5 weeks.

Expected Output: A physical, interactive construction that you can show off in Week 5 around the dorms and/or at the project fair.

Difficulty: 🌶🌶–🌶🌶🌶🌶

Prerequisites: None. In particular, you can do this project even if you don't have programming experience.

Travis's Projects

Reimagining school (Travis)

Description: How would you redesign school if you could start from scratch? This project will give you a chance to answer that question in a detailed way. We'll use some articles by some academics and education journalists to jumpstart a discussion about what school is for and what it could be. After that, we'll design a new school system, from the high-level principles to how it might actually be implemented.

Everyone has ideas and complaints about school; this is a chance to suggest something better!

Structure: We'll spend about a week and a half reading and discussing articles, and then we'll start trying to build a school system from scratch

Expected Input: At the beginning, several hours of reading. After that, lots of daydreaming about what school could be and how to implement your pie-in-the-sky ideas

Expected Output: A detailed plan for your dream school system!

Difficulty: 🌶–🌶🌶🌶

Prerequisites: None

Write a fairy tale (Travis)

Description: Once upon a time, in a campus tucked into a faraway green land, lightning struck. It was, of course, a dark and stormy night. Perceptible only in the flashes of lightning, a young camper was making their way through the lashing rain to the stronghold of Valcour Hall. There lied safety, protection granted by the hallowed Staff of Wisdom, but only if the Sign-in Ritual could be completed by 10pm. Otherwise, this young camper would not only be running from their enemies, but also from the Workshift Warriors…

So our young camper fought valiantly against the winds that whipped wildly against the willows around them. They staggered toward the Behatted Staff to complete the ritual, the possibility of safety and the potential of its absence coupling with exhaustion to make even the simple act of walking nigh impossible. The pen drops from their hand as they complete the last line of their logbook and they hear, as though from far, far away, the Sigil of Safety: "You are signed in." They collapse on the floor, fully spent, and just as their eyes begin to close, a packet of blurbs flutters to the floor beside them, where they glance upon this blurb and realize this is only the beginning of the story…

Structure: Before our first meeting, we'll read some fairy tales. In our first meeting, we'll the unique qualities of the fairy tale to kickstart our writing process. Then we'll start writing, we'll meet to brainstorm and provide feedback on what we've written.

Expected Input: about 2 hours in the first week, 1+ hours/week following

Expected Output: A fairy tale written by you!

Difficulty: 🌶

Prerequisites: A desire to write!

The staff's Projects

Teach a class (The staff)

Description: Do all of the wonderful classes you've been taking at Mathcamp make you want to try teaching one of your own? Well, you can! If you sign up to do a teaching project, you'll work with a member of the teaching staff to put together a one-day class for Week 5. This could be any of us, so ask around! Even if one of us isn't the best choice, we can probably tell you who is.

Structure: Meet periodically with whoever is overseeing your teaching project to talk about how to structure the class and give practice talks.

Expected Input: This will be a lot of work, probably more than you think! Expect to have lots of meetings over the course of camp and practice your class multiple times.

Expected Output: If the member of the staff overseeing your project thinks you're ready in time, your class will run as part of the Week 5 schedule.

Difficulty: 🌶🌶🌶🌶

Prerequisites: Enthusiasm and a willingness to work hard on presenting your ideas well.