PROJECT DESCRIPTIONS — MATHCAMP 2022

Contents

Aaron's Projects

Model theory (Aaron)

Description: At Mathcamp, we encounter loads of different mathematical widgets. There are groups, graphs, posets, tosets, rings, fields, vector spaces, and more. That's a lot to keep track of, but with model theory, we can view all of these as examples of the same phenomenon. We even give general definitions of "mathematical structures," "axiom systems," and "proofs," which allow us to take a unique birds-eye perspective on math.

If you want to read about how to tie these things together, or you took my class on Model Theory last year and want to read on beyond where we left off (or fill in the details we had to skip), join me! We will read part of *Model Theory* by David Marker, possibly with some other sources thrown in, depending on what interests you.

The end goal of the project is flexible, but one possibility is a poster explaining either a particular model-theoretic construction, or the application of model theory to a particular structure or theory.

Structure: Reading project

Expected Input: You'll spend about three hours a week reading, and we'll meet at least once a week to discuss the reading, and determine the next week's reading.

Later in camp, we will see if there is a particular topic you'd like to present in depth at the poster session.

Expected Output: Deeper knowledge of this side of math, and perhaps a poster explaining some of it to your fellow campers.

Difficulty: 🌶🌶🌶🌶

Prerequisites: It depends on who's interested. If we start at the beginning of the story, you should be comfortable with at least one of graph/group/ring theory, the more the better.

Other topics will also require some knowledge of model theory (about the level of my class last summer, but just ask me if you're not sure).

More formal proof verification in Lean (Aaron)

Description: In the Lean class, we wrote computer-verifiable proofs of basic number theory results, like the fact that there are infinitely many primes, and there's no rational square root of 2.

Given more time, we can go further, and prove more difficult results in Lean. In this project, we will choose one of your favorite pieces of math, and prove things about it in Lean.

Depending on how far we get, we may even be able to contribute to mathlib, the online library of math in Lean. (Campers have done this each of the past two summers!)

Structure: You code proofs in Lean, mostly on your own time. I will provide hints and direction in weekly meetings and/or Slack messages.

Expected Input: Several hours a week, spent mostly coding proofs in Lean.

Expected Output: A computer-verifiable proof of a result you like. Possibly a mathlib submission.

Difficulty: 🌶🌶🌶🌶

Prerequisites: Experience with Lean, presumably from the Week 1 class. Having a discussion with Aaron to generate topic ideas.

Aaron and Marisa's Projects

Songwriting crash course (Aaron and Marisa)

Description: Do you like playing or singing music?

Do you want to try writing your own songs, but you don't know where to start? In this project, we'll teach you the basic ideas you need to start, and give you the feedback you need to finish an original song.

Structure: Weekly meetings. These will start out with more instruction, but will give way to active advice and feedback. In between meetings, you'll work on your song(s) individually.

Expected Input: About 2 hours of independent creative writing and practice, and one hour of meeting a week.

Expected Output: An original song (which you can play at Song Swap or the Talent Show!)

Difficulty: 🌶🌶

Prerequisites: You should be comfortable either singing or playing an instrument available at camp, and be willing to pick up a little bit of either piano or guitar to write chord progressions.

Ania and Zoe's Projects

Measuring Covid risks in various indoors spaces (Ania and Zoe)

Description: Would you like to know how safe is going to classes? Using a shared bathroom? Going swimming? In this project, we will try to gather some quantitative data on rooms' ventilation (using a CO2-meter), analyze them, compare with existing guidelines and read some papers to explain our findings! We can explore the impact of fans, open windows, room size, and anything else that comes to your mind.

Structure: The project will consist of several parts: collecting data from a CO2-meter, making room measurements, doing some data analysis, comparing our data with data from this MIT app: `https://indoor-covid-safety.herokuapp.com/`, and reading relevant papers.

Expected Input: Few hours a week, but very flexible!

Expected Output: Pretty visualization of our findings with explanations! Ideally we'll be able to tell what's the impact of fans, open windows, room size, (. . .).

Difficulty: 🌶–🌶🌶

Prerequisites: None

Arya's Projects

Casting new hexes (Arya)

Description: The game of Hex (although no longer in print) is objectively better than tic-tac-toe—we still have two players drawing Xs and Os on hexagonal tiles in a way to connect their tiles in a path while blocking the opponent. However, the game can never be tied, and at least one player leaves the game sad. Surprisingly, the fact that the game of Hex cannot be tied is useful in proving a big bunch of topological results in TOTALLY non-obvious ways! Things seem dandy, but what if we tile the plane with shapes that are not hexagons, and play the same game? Moreover, everyone knows that hexagons are shapes that are CLEARLY better suited to be hyperbolic; wanna try playing Hex in negatively curved ways? Let's play!

Structure: Weekly meetings + thinking + reading.

Expected Input: Approximately 2 hours per week (if not more), including the meetings.

Expected Output: A poster for the fair, maybe some research (depending on how things go).

Difficulty: 🌶🌶

Prerequisites: Knowledge of hyperbolic geometry is welcome, but can be developed along the way.

Counting extremal fundamental domains for surfaces (Arya)

Description: Start with an even-sided polygon, and glue pairs of sides together. What can you say about the resulting surface? For example, starting with a square, if we glue a pair of opposite sides together, we get a cylinder. Glue both the pairs of opposite sides together, and you get a torus (the surface of a donut). Are there other polygons (with some pairs of sides glued) that can give you a cylinder? A torus? What about other surfaces?

In this project, we shall explore ways to glue polygons to obtain surfaces. Specifically, we shall study the possible number of sides a polygon can have to obtain a fixed surface, the number of different ways you can glue an n-gon to obtain this surface, and how this relates to the geometry on the surface.

Structure: Campers work in a group, and meet with me twice a week (for 30 minutes each) to discuss progress.

Expected Input: 2 hours per week of exploration and reading, and attending the meetings.

Expected Output: Something for the project fair, but more broadly, understanding geometry and topology of surfaces.

Difficulty: 🌶🌶🌶🌶

Prerequisites: Some familiarity with the classification of surfaces, groups, and hyperbolic geometry would be useful, but we can develop any of these during the project as well.

Fishing in deeper waters (Arya)

Description: Fish is a fun(?) 6-player card game involving memory, strategy, deception and glory. The players play with a standard deck of 54 cards, divided into nine subsets. The goal of the game is for a team (of three players) to obtain 5 subsets, by "fishing" for cards. For a detailed set of rules and gameplay, look at `http://cfish.herokuapp.com`. The focus of this project is to study fish from a game-theoretic perspective, as well as generate variants of fish that make the game harder (cus, ya know, remembering 54 cards is easy). Questions to be asked: what distributions of the deck are "fair"? What are some optimal strategies to play the game? How would you code a bot to play the game? How can we make an "interesting" variant of the game with multiple decks of cards? Sign up to brainstorm!

Structure: We meet weekly, play fish and have thoughtful conversations.

Expected Input: About an hour weekly, brainstorming ideas.

Expected Output: A compilation of several interesting variants of fish to make the game better (if possible)

Difficulty: 🌶

Prerequisites: None

Let's solve Onitama! (Arya)

Description: Onitama is a two player, perfect information board game (similar to chess) designed by Shimpei Sato. On a 5×5 board, both players start with five pawns on their side, with the main pawn in the middle. Both players have a choice of two moves they can make on their turn, and there is a fifth move card open. The moves keep being rotated between the players as the game goes on, so both players can easily calculate the possible moves their opponent can make. Taking the opponent's main pawn, or moving your main pawn into your opponent's main pawn's starting space, wins you the game. For a free online version, see `https://onitama.app/#/`.

In this project, we shall play a bunch of Onitama, discuss possible winning and losing strategies in certain scenarios, and talk about which move cards are fair. We could further discuss and develop other variants of Onitama. This project could serve as an introduction to two-player games, via working through this particular game.

Structure: Meetings twice a week for 30 mins each.

Expected Input: Attend weekly meetings, play loads of Onitama with other participants, make (and maybe prove) conjectures.

Expected Output: A poster for the project fair. This would be original math, so writing a paper in the future wouldn't be too unreasonable.

Difficulty: 🌶🌶

Prerequisites: None

Origami with surfaces (Arya)

Description: Topologically, one can express a torus as a square with opposite sides identified. However, a simplicial structure on the torus is not quite simple—where we build a torus by gluing triangles together, with any two triangles sharing at most an edge. What is the minimum number of triangles required to get such a triangulation for the torus? What about a cylinder? A Klein bottle? Other surfaces? In this project, we shall study simplicial structures on surfaces mathematically and physically. We shall try to build various surfaces by gluing the fewest possible triangles together.

Structure: First week would be going over classification of surfaces and Euler characteristic, and trying to glue triangles together to get a sphere, a cylinder and a torus. The following weeks would be trying to play around with triangulations, constructing the optimal triangulation for a torus, and building some models for higher genus surfaces.

Expected Input: Roughly an hour of work per week, divided into math and origami. An hour of meeting each week to discuss.

Expected Output: Some origami models for the project fair, but more broadly, understanding simplicial complexes.

Difficulty: 🌶🌶

Prerequisites: Some familiarity with classification of surfaces, but that can be built during the class.

Assaf's Projects

EDI: literature review (Assaf)

Description: You're the chair of a math department/high-school principal/running a math circle/doing math at a Mathcamp, and you've decided that it's time to start thinking about issues of equity, diversity, and inclusion (EDI) with an organized perspective. In this project, we will read some studies and surveys related to EDI, and discuss their merits.

Structure: Reading and discussion

Expected Input: 2–3 hours a week for reading, 1 hour discussion

Expected Output: Poster presentation and a written summary of best practices

Difficulty: 🌶

Prerequisites: None

It's all about perspective (Assaf)

Description: There's a really neat book by Marc Frantz and Annalisa Crannell about the mathematics of perspective drawing (`http://mathserver.neu.edu/~eigen/U220/Viewpoints-5-19-07.pdf`). If you're artistically-inclined, and are interested in reading this book, I'd like to read it together!

Structure: Weekly reading and drawing what we learn about

Expected Input: 2–3 hours per week + 1 hour/week meeting

Expected Output: Poster with pretty pictures and explanations

Difficulty: 🌶🌶

Prerequisites: Drawing skills **not** required

Straight lines on the tetrahedron (Assaf)

Description: An ant walks along the surface of a tetrahedron, starting in the middle of a face, and eventually gets to an edge. She decides that "going straight" along that edge means "preserving the angle with the edge, but continuing on the other side." Naturally, some questions arise: will she ever return to where she started? For any point, is there always a direction to return to where you start? Are there any periodic paths? Can any path start and end on a vertex? What if instead of a tetrahedron we use a dodecahedron? All of these questions, and more, will be explored in this project!

Structure: Meetings, plus presentations of finding to group

Expected Input: ∼ 4 hours per week

Expected Output: Poster!

Difficulty: 🌶🌶🌶

Prerequisites: None, but Teichmüller theory of torus may be useful

Teichmüller theory (Assaf)

Description: So. . . you've learned about the Teichmüller space of the torus, and now you must be wondering: can we do the same thing for other surfaces? The answer is yes, but the picture is a lot more complex (and interesting!). If there's time, we'll get to defining some interesting metrics on Teichmüller space, and try to solve some open problems about them.

Structure: Reading project

Expected Input: 3 hours a week or so

Difficulty: 🌶🌶🌶🌶

Prerequisites: Hyperbolic geometry, Teichmüller theory of the torus

Write the Mathcamp manifesto (Assaf)

Description: "A spectre is haunting mathematics—the spectre of Mathcamp. All the powers of old Math have entered into a holy alliance to exorcise this spectre: Analyst and Algebraist, Geometer and Set Theorist, Topological Radicals and physicist police-spies."

Mathcamp has existed for nearly 30 years. It has an anthem, a people, a logo, but no manifesto. In this project, we will write this manifesto.

Structure: We will read various manifestos to get the feel of them, and then write one for Mathcamp

Expected Input: A few meetings over the course of camp (maybe once a week) to read manifestos and think about manifesto writing

Expected Output: The Mathcamp Manifesto

Difficulty: 🌶

Prerequisites: None

Zine-ify a class (Assaf)

Description: "A zine (/zin/ ZEEN; short for magazine or fanzine) is a small-circulation self-published work of original or appropriated texts and images, usually reproduced via a copy machine." Small tid-bits of math are perfect contents for a zine! This project is about turning Mathcamp classes into zines to print, share, and enjoy! For some examples of math-zines, see: `https://www.math.toronto.edu/~safibn/zines.html`.

Structure: Loose meetings

Expected Input: About an hour a week

Expected Output: A zine

Difficulty: 🌶

Prerequisites: Taken at least one class at Mathcamp

Assaf and Marisa's Projects

Singing proofs to ancient tunes (Assaf and Marisa)

Description: The art of cantillation dates back to the 10th century CE, and it is used to read every verse in the Hebrew Bible with a musical chant for Jewish liturgical purposes—but rather than sheet music, it uses diacritical marks on each word to indicate the melody. The choice of each diacritical mark is not just melodic, but actually formulaic and (mostly) predictable: it boils down to a thoughtful diagramming of a sentence in a tree structure. (First, split the verse in half; then identify the dependency of words in each clause. . .) The diacritical marks then essentially become not just the melody, but also the punctuation: they tell you where the commas belong, and where the periods and question marks are. (This is particularly useful, because the Bible was written originally by scribes without breaks between sentences. Or punctuation at all.)

Now, there's no particular reason why this system of musical punctuation isn't applied to other texts. Say, for example. . . a proof that the square root of two is irrational.

In this project, you will learn about the system of cantillation, and apply it to your favorite math proofs. Despite the origin story, no interaction with the Bible is involved in this project: we'll just be learning about the musical notation and how it relates to sentence diagramming, and then singing math proofs to each other in a highly technical way. (The singing is very loose; you do not need any musical background. Being basically able to carry a tune definitely suffices.)

Structure: Weekly meetings, and some between-meeting practicing.

Expected Input: You bring the proofs that you want to work with, and you'll need to practice to learn the cantillation system and the sentence diagramming.

Expected Output: A cool-looking annotated proof, and the option of singing it to your friends.

Difficulty: 🌶

Prerequisites: None

Ben's Projects

Learn a language (on Duolingo?) (Ben)

Description: Have you ever wanted to learn another language? While Mathcamp isn't long enough to fully learn a language, it has a lot of awesome campers willing to try new things! This project will give you an opportunity to start learning a language with other campers and have conversations of the form "the dog has the table" or "where is the chair." We may invite a small green owl to the proceedings, as well.

Languages that I'm slightly more than ignorant of include Latin, German, and (maybe) Russian, but I'm happy to try others too!

Structure: Students would practice the language on their own

Expected Input: About 15 minutes to half an hour of time practicing the language a day and a meeting or so a week

Expected Output: Ability to say such sentences as "this cat is green" and "where is the library" in a new language

Difficulty: 🌶

Prerequisites: Ability to speak at least one language (Your native language counts. So does English.)

Ben and Zoe's Projects

Fractal baking (Ben and Zoe)

Description: Fractals are often very pretty and nice to look at. Baking is the art of making bread and bread-like products such as cakes, pies, etc. In this project, we will try to bake things and then make fractal-esque shapes! In the process, we'll talk more about what makes a fractal a fractal.

Structure: We'll meet a few times a week

Expected Input: Time, effort at baking

Expected Output: Delicious food and fun pictures

Difficulty: 🌶

Prerequisites: An enthusiasm for baking and a willingness to think about fractals!

Charlotte's Projects

Measure theory (Charlotte)

Description: In this project you will be introduced to the wonderful world of measure theory. We know how to measure the area of some basic shapes in the plane, but measure theory will allow us to measure the size of many many more subsets of $\mathbb{R}^2$. It also leads to a more general and elegant form of Riemann integration, called Lebesgue integration. We'll read part of Terrence Tao's *An Introduction to Measure Theory*, and learn the fundamentals of Lebesgue measure and Lebesgue integration.

Structure: You'll be assigned sections of the text to read each week, and we'll meet weekly to discuss the concepts.

Expected Input: This could range from 5–8 hours per week.

Expected Output: Lots of knowledge

Difficulty: 🌶🌶🌶–🌶🌶🌶🌶

Prerequisites: Have taken an intro to real analysis course before

Charlotte and Natali's Projects

Genius (Charlotte and Natali)

Description: In this project we will introduce you to the greatest game show ever, the Genius. Each episode of the Genius centers around a game involving math, strategy, and social play. In this project, we'll organize and play various Genius games, and discuss our strategies.

Here's an example of a game we could play: In "Food Chain" each player gets an animal role. Roles are roughly split into predators and prey. Each player has a different win condition and a different capability; for instance, the lion wins if it eats a prey every round. If a predator tries to eat the snake then the predator dies. Players choose which habitat (e.g., field, river) they would like to go to every round and can only eat animals in the same habitat.

Here's another example: "Minus Auction" is a sort of (reverse?) auction. The goal is to have the highest point total. Players are given chips, each worth $+1$. Suppose that the number -17 is put up to bid. Each player can pass on the number by throwing a chip in the pool. When one player decides to take -17, they receive this number (worth -17 points) but also receive all chips in the pool. Also, if a player obtains a streak of negative numbers, say -17, -18, -19, only the largest number (-17) contributes to their point tally.

Structure: We will organize and play about 1 game per week.

Expected Input: Say 4ish hours per week.

Expected Output: FUN!

Difficulty: 🌶
Prerequisites: None

Emily's Projects

Poset topology and simplicial complexes (Emily)

Description: This is a reading project on selected topics related to poset topology and simplicial complexes. We will start with studying the relationships between posets and complexes, and then jump around to selected topics of interest, such as shellability, Cohen–Macaulay complexes, and/or group actions on posets.

I plan for us to use parts of Michelle Wach's poset topology notes, found here:
`https://arxiv.org/abs/math/0602226`

Structure: Reading and working on examples on your own, meeting 1–2 times per week to share solutions and discuss parts of the reading.

Expected Input: Anywhere from 3–6 hours per week, depending on the amount of time you have/are willing to put in.

Expected Output: Knowledge! We can also explore presenting at the project fair.

Difficulty: 🌶🌶🌶–🌶🌶🌶🌶

Prerequisites: Depending on the exact direction we go, linear algebra, group theory, and ring theory would be beneficial. You can/should still talk to me even if you don't have these!

Writing music with the Tonnetz (Emily)

Description: The Tonnetz is a triangular lattice which visualizes triads, consonance and dissonance, and many chord progressions. In particular, the vertices of the lattice are the 12 notes of the chromatic scale, and they are arranged so that triangles represent major/minor triads. The lattice is self-repeating in all directions, so it can wrapped around a torus.

The plan for this project is to use the Tonnetz to write our own music! For example, paths throughout the lattice correspond to chord progressions, so we can choose an interesting path to represent our chords (you can decide what "interesting" means). We could also choose interesting patterns of notes on the Tonnetz to develop a melody. This will be a chance to get creative!

Structure: Meeting 1–2 times per week. We will spend the first meeting or two studying the Tonnetz (if you didn't take my class), and I will build a basic example of how we can use this to create our own music. In future meetings, we will look at your progress, resolve any issues you are having, and brainstorm ideas for what to do next. We can also explore other ways of incorporating math into your piece(s).

Expected Input: At least 2 hours per week outside of meetings.

Expected Output: Piece(s) of music that you have written. This could take many forms; you could choose to write several short pieces that each represent one concept on the Tonnetz, or one piece which combines many ideas. Preferably, this will be written using some music composition software (like Musescore). You may choose to present at the project fair.

Difficulty: 🌶🌶

Prerequisites: W1 Geometry of Music and/or recent experience with music theory (playing an instrument counts!).

Eric's Projects

Aluminum foil dorodango (Eric)

Description: Dorodango is an art form that involves making spheres out of mud/clay/dirt and polishing them until they are very shiny. We'll do something very similar with aluminum foil! That's the whole project: take a roll of aluminum foil and hammer it/sand it/polish it until it's a shiny fist-sized sphere. I'll provide all the materials and tools we need.

Structure: There's several tutorials on youtube, so I think we'll want to watch those and distill the best technique before diving in ourselves. The bulk of the project will be relatively mindless manual labour.

Expected Input: Several hours (I'm expecting ∼5 hours for a single sphere) of manual labour, spread over a few sessions.

Expected Output: Aluminum foil dorodango souvenir. Depending on how long it ends up taking us to make them we may iterate on our technique with further attempts.

Difficulty: 🌶

Prerequisites: None

Haiku and cubic surfaces (Eric)

Description: The goal of this project is to exploit a numerical coincidence to produce a very cool piece of math-related art. It's a very neat fact in algebraic geometry that any (smooth) cubic surface (basically the set of solutions to a degree 3 polynomial in 3 variables) contains exactly 27 lines. For a particular cubic surface (the Clebsch cubic) there are 10 Eckardt points: points where 3 of these 27 lines meet. A haiku is a poem with 3 lines, which have 5, 7, 5 syllables each. I think there's a very cool piece of art waiting to be made where each line on the surface gets a 5 or 7 syllable line of poetry put on it, in such a way that you can read a haiku off of each Eckartdt point! Let's try to make something like that!

Structure: We'd learn a bit about the math of cubic surfaces (though full proofs are well beyond the scope of the project), and then have fun playing around with how to make interestingly constrained poetry around them!

Expected Input: At most 4 hours total at TAU to learn what we'd need about cubic surfaces, and then as much time as you want playing around and writing poetry.

Expected Output: A cool piece of mathematically motivated poetry to show off at the project fair!

Difficulty: 🌶🌶

Prerequisites: None

Learn programming through Advent of Code (Eric)

Description: Advent of Code[1] is a yearly programming challenge in the form of an advent calendar that is an excellent way to learn programming and computer science concepts. Each year is themed around a certain whimsical Christmas-themed goal (rescuing Santa from outer space, recovering the keys to Santa's sleigh from the bottom of the ocean. . .); the puzzles start out being very simple, solvable in one line of code, and end up being quite complex (the last puzzle from 2019 is a full-fledged text-based adventure game that you have to write a virtual machine to run and then beat).

We'll try to solve one year of puzzles, learning whatever coding tools and CS concepts we need along the way. I'm happy to supervise this in any language, and if you don't know anything about programming already I'll help you get set up with using one!

[1] `https://adventofcode.com`

Structure: Several hours of TAU time devoted to solving problems every week; we might have very small talks during TAU if there's a particularly useful CS concept for solving the problem we're attempting.

Expected Input: At least 3 TAU hours per week devoted to solving Advent of Code problems. This project can be as spicy as you want; there's 7 years of advent calendars so I can pretty much guarantee you won't run of challenges no matter how fast you work.

Expected Output: Solutions to advent of code puzzles, programming knowledge, possibly a project fair presentation showing off our best solutions.

Difficulty: 🌶🌶–🌶🌶🌶🌶

Prerequisites: Nothing! In particular you don't need to know anything about programming already, the point of this project is to learn by doing.

Mapping Romeo and/or Juliet (Eric)

Description: Romeo and/or Juliet is a wonderful book where you play through Romeo and Juliet in a "choose your own adventure" style; you can choose to follow the original story, but you can also choose to take it in some crazy new directions (you can choose to have Juliet ditch Romeo to become a pirate captain! this book has unlockable characters! wild stuff like this awaits). The project is to read the book, and while doing so create a graph of all the choices you can make. It'd be super neat to see the whole intricate structure of this book laid out after having explored it.

Structure: This is a super chill project! Read a cool book and make some neat visualizations of it.

Expected Input: About an hour meeting every week to talk about progress and neat things we might do, along with a fair amount of solo reading and information gathering time (the book is 400 pages but it reads very fast).

Expected Output: It'd be great to show off the graph and whatever other visualizations we uncover at the project fair! We could also run some dramatic readings or other events for other campers.

Difficulty: 🌶

Prerequisites: None

Propose a new emoji (Eric)

Description: Did you know that anyone can propose new emoji[2]? And that if your emoji gets selected then your idea is forever immortalized by like every phone manufacturer and social network? AND it doesn't even seem to be that hard. So the project is: do a bit of research on what is involved in putting together a proposal, brainstorm a new emoji, make a proposal, submit it! With a bit of care to submitting different emoji, several groups of campers could do this.

Structure: A few discussion meetings to brainstorm what emoji we'd like to submit, and then structured work either during TAU or in the evenings to put together the proposal.

Expected Input: Read and understand the emoji submission standards, work with other campers on putting together a proposal.

Expected Output: Submit a proposal for a new emoji!

Difficulty: 🌶🌶

Prerequisites: None

[2] `https://unicode.org/emoji/proposals.html`

Gloria and Linus's Projects

Resolve a square dance (Gloria and Linus)

Description: "Square dancing" is a mix of logic and dancing. Dancers are given instructions like "Swing thru," which means to rotate 180 degrees around your right hand (if you're holding right hands with someone) and then do the same for your left hand.

Square dancing usually requires 8 dancers, but because every reachable position is 180-degree symmetric around the origin, you can dance with 4 instead by applying the complex map $z \mapsto \sqrt{z}$. But anyway…

In this project you will learn some square dance calls, and then figure out a method for "resolving" the dance (getting all the dancers back to their original position).

Expected Input: Each week we'll spend a couple hours at TAU learning/practicing square dance calls. Then, writing a good/legal square dance and resolving it is up to you.

Expected Output: Call a square dance!

Difficulty: 🌶🌶

Prerequisites: None

Gloria and others's Projects

Mathematical yarn and quilts (Gloria and others)

Description: This is not a project in the same sense as the others: in particular, you don't need to sign up for it, and you can participate even if you are participating in another project!

We have a wonderful crafts lounge in Foss 116 with many materials for making beautiful and interesting mathematical objects. You can knit or crochet things as simple as knitted hyperbolic planes or as time-consuming as a Klein bottle hat, or Möbius strips, or many other objects. You could even design and make a quilt! Ask your instructors for more ideas, or look here[3]. 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 Gloria or another staff member.

Expected Input: Between one hour and lots

Expected Output: Fun objects

Difficulty: 🌶

Prerequisites: None

Kayla's Projects

Discrete Morse theory (Kayla)

Description: Go through one of my favorite pieces of math writing: *A User's Guide to Discrete Morse Theory*, which discusses a combinatorial version of Morse theory. We will spend time building up CW complexes with a combinatorial recipe that essentially just assigns numbers to simplices.

Structure: Meet once a week in a small group where campers rotate giving small talks about the reading they did. We will also discuss examples outside the text and work together through some exercises.

Expected Input: ∼2 hours of reading a week

Expected Output: A new love for combinatorial topology

Difficulty: 🌶🌶

[3] `https://tinyurl.com/2t4f2n2w`

Prerequisites: Some exposure to topology of $\mathbb{R}^n$ would be great

Dynkin diagram and root systems (Kayla)

Description: Come learn about why Dynkin diagrams are amazing and classify all sort of rich mathematics. In this project, we will be learning what a root system is and classifying them. I would also be happy to chat about other mathematical structures that are classified by these diagrams if there is interest :)

Structure: Reading something and chatting!

Expected Input: As much reading as you want to do

Expected Output: Understanding why people love these diagrams!

Difficulty: 🌶🌶–🌶🌶🌶🌶

Prerequisites: None

Is it cake? Is it a quiver rep?? (Kayla)

Description: Why do we like cake? It's sweet, delicious, nutritious and probably means celebration. Why do we like quivers? They're visual, succinct ways to express a lot of information, and allow us to find combinatorial ways to understand deep algebraic topics. In this project, we will see how representations of finite dimensional algebras (complicated and spooky algebraic objects) are really just quiver representations in disguise. We will define and motivate quiver representations, path algebras, and the big punchline we will aim to understand the statement of is the equivalence of categories between representations of certain algebras and bound quiver representations.

Structure: Reading and discussion

Expected Input: 1–2 hours a week

Expected Output: A new appreciation for directed graphs

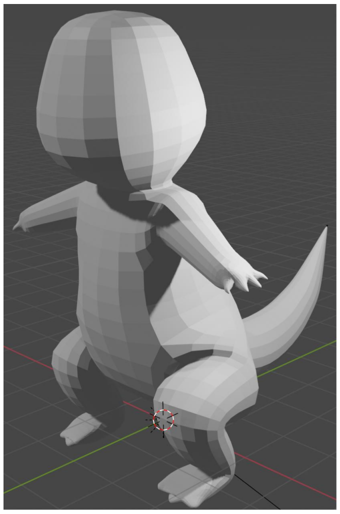
Difficulty: 🌶🌶🌶🌶

Prerequisites: Linear algebra (understanding vector spaces and linear transformations), group theory (homomorphisms)

Linus's Projects

Blender (Linus)

Description: Make 3D models like chess pieces or Pachirisu. Make one of those neat physics animations where the blue blocks fall off the table and the red blocks stay on. Learn about optics principles like Fresnel and dielectric surfaces. Experiment with shader code to create pretty and/or mathy visual effects. This project will involve technical thinking—e.g. to figure out the best way to sculpt a sword using basic operations, or, to figure out the sequence of operations to apply to a vector to make a polka dot pattern. But in the end, it is more creativity than math.



Expected Input: You get out what you put in. Expect several hours of ramping up before you can comfortably create shapes as complex as "my keyboard" or "a pawn." From there you can prototype a bunch of demos that play with different features of Blender. Or you can go deep and spend 8–∞ hours modeling and combing the hair on an Eevee.

Expected Output: Some renders or a scene to show at the Project Fair.

Difficulty: 🌶🌶🌶🌶

Prerequisites: Have a decent laptop. I recommend you try installing and opening Blender before project selection. Explicitly not a prerequisite is art skill: IMO, modeling and sculpting in 3D is, at least at the beginner level, orthogonal to pencil-and-paper drawing skill.

PCA map of food (Linus)

Description: Principal Component Analysis is a technique to compress and find correlations in high-dimensional data. Famously, applying PCA to human genomes in Europe generates an approximate map of Europe. Let's survey a lot of Mathcampers about a lot of different food preferences, and use PCA (and maybe some other algorithms) to generate a map of food.

Structure: First we'll figure out exactly what to put on the survey. Then we'll conduct the survey. Then we'll analyze the data, and solve problems like "What do we do with campers who ranked everything 10/10?" and "What do we do with foods like Gochujang where only 8 campers know what it is?"

Expected Input: Be willing to ask a lot of Mathcampers to fill out a survey.

Expected Output: A map of food.

Difficulty: 🌶–🌶🌶

Prerequisites: Linear algebra useful but not required.

Prove stuff NP-complete (Linus)

Description: Prove that your favorite logic puzzles, or my favorite logic puzzles, are NP-complete.

Expected Input: Depending on the difficulty, expect it to take a few hours to prove something is NP-complete. Plus a couple hours to learn if you don't already know what NP-complete means.

Expected Output: Maybe a fun tool to convert one problem to another?

Difficulty: 🌶🌶

Prerequisites: It would help to already know what NP-complete means, but I can also explain it to you.

Misha's Projects

Design a Dominion expansion (Misha)

Description: Dominion is a set-building card game that, since its release in 2008, has seen 14 different expansions with more cards.

In this project, you'll design your own expansion for this card game. This includes:

- Picking a new theme for the expansion (ninjas? dinosaurs? mathematicians?)
- Coming up with lots of cards in that theme that have new exciting mechanics or are just fun to play with.
- Doing lots of play-testing to determine how good the new cards are and how they should be rebalanced.
- Ruthlessly cutting cards that don't work.

Structure: We'll start with a planning meeting, then I'll check in 1–2 times a week on your progress (or whenever you feel like you need advice).

Expected Input: Lots of play-testing.

Expected Output: At least a small expansion (12–13 new cards). If you have the time and energy, maybe a large expansion (25 new cards).

Difficulty: 🌶🌶

Prerequisites: You must know the rules for Dominion (at least the base game).

Kromatikus Gráfokról (Misha)

Description: Graph theory is best theory, and the Hungarian language is the weirdest language.

Let's read a graph theory paper by Pál Erdős and András Hajnal in Hungarian! I challenge you to decipher as much of it as possible without consulting a dictionary (or Google Translate).

In the process, you might learn something about graph coloring and/or about Hungarian.

Structure: I will print out the paper for you (it's about 3 pages long), and I can help a little with the stranger aspects of Hungarian grammar.

Expected Input: You read the paper and maybe make notes on what you think parts of it mean.

If you get through it quickly and want more, I can find more papers in other languages.

Expected Output: Maybe you'll be able to explain what the paper says to other people!

Difficulty: 🌶🌶🌶🌶

Prerequisites: You must not already know Hungarian, or else this will be too easy. Some graph theory background might help, but I'm not sure it will help enough.

Open problems in cartography (Misha)

Description: You may have heard of the four-color theorem. Something something maps something something adjacent countries.

We can ask more questions about how many colors are needed to color stranger maps than usual. For example:

- What if every country can have two disconnected pieces that must still be given the same color?
- What if the shape of countries is somehow restricted?
- What if every country has a colony on the moon, and we want to color both planets in coordination? (This one is an open problem!)

We can ask further questions, maybe even ones that nobody has ever thought about before, by building on the ideas in my upcoming week 2 colloquium.

Structure: You can come up with your own problems in cartography and answer them!

If you want suggested problems, I can suggest problems.

If you want to read about what other people have done with those problems, I can suggest some background reading.

Expected Input: Meetings 1–2 times a week to describe progress.

Expected Output: You will definitely prove some things!

Difficulty: 🌶🌶

Prerequisites: Some knowledge of graph theory might help.

Narmada's Projects

Teeny teens on trampolines (Narmada)

Description: This project is all about a fun problem. n teens are on n trampolines in a row. They want to congregate on one trampoline to throw a party. Because there's strength in numbers, k teens can jump to a trampoline at distance k. They cannot jump to an empty trampoline because there's no one to catch them. Is there a sequence of jumps to unite the teens for the party of the summer?

It turns out, we can answer this question when the n trampolines are in a row, which is a path graph. The goal of the project is to study this problem on arbitrary trees. When does a solution exist?

Structure: Meeting for 1 hour per week to construct examples and create conjectures

Expected Input: 1–2 hours per week + meeting time

Expected Output: Poster for the project fair

Difficulty: 🌶🌶

Prerequisites: Graph theory: know what a tree is

Narmada and Travis's Projects

How to right the write (Narmada and Travis)

Description: Most mathematical writing is bad. But it doesn't have to be! YOU can break the cycle. The goal of the project is to learn how to write mathematics well. We'll analyze some mathematical writing, read about good writing practices in general, and peer review each other's work.

Structure: Meeting as one group for $\sim$1 hour per week

Expected Input: Attend the weekly meetings. Campers can put in as much time practising as they want. Probably no more than 1–2 extra hours per week.

Expected Output: A short piece of mathematical writing

Difficulty: 🌶

Prerequisites: None

How to talk the talk (Narmada and Travis)

Description: Most mathematical talking is worse. But it doesn't have to be! YOU can break the cycle. The goal of the project is to learn how to give mathematical talks well. The tentative plan is to meet for 30 minutes twice a week. At the first meeting, a mentor will give an instructively bad talk for the first 10 minutes so that we can discuss how to improve it. At the second meeting, we'll work on more constructive ways to prepare good talks. The campers can give practice talks in week 4 to get feedback.

Structure: Meeting as one group for 30 minutes twice a week

Expected Input: Attend the weekly meetings. About 1 hour per week to prepare their talks.

Expected Output: A 5–10 minute talk by individuals or small groups of campers

Difficulty: 🌶

Prerequisites: None

Narmada and Yuval's Projects

Combinatorial problems and exercises (Narmada and Yuval)

Description: In the late 1970s, a brilliant 30-year-old mathematician decided to write an introductory textbook to more or less all of combinatorics, presented in the format of 607 exercises. The result is Lovász's *Combinatorial Problems and Exercises*, which remains one of the best combinatorics textbooks there is. In addition to the 607 problems, it features clever hints and beautiful solutions to each.

In this project, you will try to solve some of the problems and exercises, and in so doing, learn a bunch of cool combinatorics. There are a lot of topics in the book, so you can pick sections that seem most appealing to you, and you can work on the problems alone or as a group.

Structure: You'll work on exercises and meet with us roughly once a week to discuss

Expected Input: 1–5 hours per week

Expected Output: Solve some cool problems and learn some cool math! If you learn something you're excited about sharing, you can make a poster for the project fair.

Difficulty: 🌶🌶–🌶🌶🌶🌶

Prerequisites: None, though some exposure to graph theory or other combinatorial topics will be helpful and will allow you to get to more advanced material

Nic's Projects

Algebraic curves (Nic)

Description: Algebraic geometry is the field of mathematics devoted to the study of solution sets of polynomial equations. It's a huge field with a thriving research community, but in its modern form it also uses a huge amount of intimidating-sounding machinery, which can make it difficult to get into.

In this project, you'll read through *Algebraic Curves* by Bill Fulton, a short book that tries to give a good introduction to one corner of algebraic geometry without quite as much of the technical machinery as some other sources. You'll get a good taste for the deep connection between ring theory and geometry that makes modern algebraic geometry tick, see a proof of Bézout's famous theorem about counting the number of points of intersection of plane curves, learn about resolution of singularities, and more.

This book is short but intense. Be prepared to spend a lot of time staring at a single page! It can be found at `http://www.math.lsa.umich.edu/~wfulton/CurveBook.pdf`.

Structure: You'll read through the book either on your own or in a small group. We might supplement this with some other reading as needed. There will be weekly meetings where we'll check in and talk about any questions you might have, and you will probably meet on your own in addition to this.

Expected Input: As much effort on the reading as you have the time and inclination for.

Expected Output: A deep understanding of the geometry of algebraic curves! You might also be interested in putting together a poster to show at the project fair at the end of camp.

Difficulty: 🌶🌶🌶🌶

Prerequisites: Enough ring theory to know about quotient rings, prime and maximal ideals, and modules.

Steve's Projects

Computability theory reading course (Steve)

Description: This is basically a continuation of my computability theory class. Campers will read selections from papers and books (and do exercises) on one of a few possible topics in computability theory. The specific topic depends on camper interest; the three I'm most prepared to focus on are degree theory (no relation to topological degree theory sadly), interactions between computability theory and set theory (this one goes great with Susan's class!), and computable structure theory, but this is pretty flexible.

Structure: Reading course

Expected Input: 3 hours a week, so a bit less than a standard class

Expected Output: None (just better understanding of a topic of interest)

Difficulty: 🌶🌶🌶–🌶🌶🌶🌶

Prerequisites: Computability theory

Steve and Zack's Projects

Involutive triangle centers (Steve and Zack)

Description: It's a fun exercise to show that the orthocenter is **involutive** in the following sense: if A, B, C are vertices of a (non-right, non-degenerate) triangle with orthocenter X, then the orthocenter of $\triangle ABX$ is C, the orthocenter of $\triangle AXC$ is B, and the orthocenter of $\triangle XBC$ is A. Put another way, **iterating orthocenters is boring**. (By contrast, you can have a lot of fun iterating *circumcenters*, but that's a story for another time.)

It's natural to ask how rare, amongst the various triangle centers out there, is this involutive property. The answer seems to be "highly rare but not quite unique." Specifically, the Online Encyclopedia of Triangle Centers (:D :D :D) has about 40,000 entries, and a possibly-non-exhaustive search performed by a pseudonymous mathematician at `math.stackexchange/mathoverflow` ("Blue") determined that the involutive property holds for...three of them. See `https://mathoverflow.net/questions/407981/`.

We're interested in the question of exactly how many involutive triangle centers there are. There are a few ways to make this precise, and the question is wide open for each one.

Structure: Campers investigate an open problem via any means they deem appropriate.

Expected Input: At least two hours a week.

Expected Output: Probably nothing, but maybe a really cool result!

Difficulty: 🌶🌶🌶🌶

Prerequisites: Euclidean geometry, stubbornness; ideally some algebraic geometry or topology as well, although that's not a priori necessary. Programming (i.e. SAGE) would probably also be useful.

Travis's Projects

Create a food (Travis)

Description: Do you like food? Do you like new things? Why not try your hand at making a new food? New food is invented all the time, and *you* can join the elite ranks of the culinary innovators by creating your own. This can be as simple as smashing two known foods together in a new way or as involved as imagining a completely new dish from the ground up.

Structure: We'll hold an initial meeting to kickstart the brainstorming process, then we'll meet once or twice a week to further brainstorm, test cook, or problem solve draft recipes.

Expected Input: 1–1.5 hours per week brainstorming foods/recipes or test cooking

Expected Output: A new food and a recipe to create it

Difficulty: 🌶–🌶🌶

Prerequisites: Know what food is

How complex is baby shark? (Travis)

Description: In 1984, Donald Knuth, a titan of computer science, wrote an article that analyzed the complexity of songs. It's a highly frivolous article, and a great read. In this project, we'll read the paper and see what Knuth meant by the complexity of a song, and then we'll get down to business; just because Knuth meant it as a joke doesn't mean we can't take it seriously! We'll pick a few songs (e.g., baby shark) and analyze them. Or, for a more ambitious project, you can pick something besides songs, define a notion of complexity on them, and analyze several examples.

Structure: Our first meeting will cover any necessary background. We'll spend 1–2 weeks reading the paper and the remaining time analyzing songs, probably meeting about twice per week for about 30 minutes to discuss progress, answer questions, and decide where to go next.

Expected Input: 1–2 hours per week independent work, about 1 hour per week meeting

Expected Output: A poster for the project fair

Difficulty: 🌶🌶

Prerequisites: None

Introduction to advanced intermediate LaTeX design (Travis)

Description: Have you ever wanted to customize the format of a PDF and been stymied by LaTeX's byzantine syntax and structure? Or seen a fancy figure in a LaTeX document and thought, "I wish I could make that"? Wish no longer! LaTeX is basically infinitely customizable, if you know how to navigate the syntax. This project can be as ambitious as you want: anything from learning to draw some figures using TikZ to learning how to customize the entire layout of the document.

Structure: An initial meeting to decide what you want out of the project, then weekly meetings to discuss how to achieve this, overcome difficulties you run into, etc.

Expected Input: 1–2 hours per week TeXing and reading LaTeX guides; more is fine—you'll get out what you put in.

Expected Output: A set of nicely TeXed notes from a class, or a nice default style file, or a collection of fun figures, or . . .

Difficulty: 🌶🌶

Prerequisites: Basic facility with LaTeX

More symbolic dynamics (Travis)

Description: Did you *love* learning about symbolic dynamics in week one? Are you yearning to learn more? Fret not, my new symbolic dynamics acolyte, for this is exactly the opportunity you're searching for.

Structure: We'll use the first meeting to decide a loose structure to the rest of the project, based on your background and interest; then the work will mostly consist of independent readings with one to two weekly meetings to check in and talk about what you've learned and the questions you have.

Expected Input: Somewhere between 1 and 4 hours of independent reading (depending on how much work you want to put in) and one or two 30 minute meetings per week

Expected Output: Knowledge. (And a poster, but only if you want to make one.)

Difficulty: 🌶🌶–🌶🌶🌶

Prerequisites: Have taken The Mathematics of forbidden words in week one. There are many interesting connections between linear algebra and symbolic dynamics, so if you took my week one class, this increases the number of directions that this project can take. (But it's certainly not necessary!) The same goes for analysis.

Viv's Projects

Analytic number theory through numerics (Viv)

Description: Analytic number theory is filled with a lot of hard questions that we don't know how to answer. Questions like, how well can we estimate the number of prime numbers that are less than x? How many twin primes are there? How much do primes cluster and spread out—that is to say, how frequently are there very big and very small gaps between primes? How do partial sums of multiplicative functions behave, or sums of multiplicative functions in short intervals? And so on.

This project will be "no thoughts, just vibes," where by "thoughts" I mean proofs, and by "vibes," I mean computer experiments. That is to say, we'll build an understanding of what we think the right answers should be to many of these questions by seeing what our computer tells us. We'll start with getting an understanding for what the statement of the prime number theorem should be before moving on to error terms in the prime number theorem and then onto different pastures. We'll explore ideas like asymptotic reasoning, numerical fits, Euler products and contributions from different primes, and what happens when you're working with a function that your computer can't tell isn't constant (such as $\log\log x$.)

Structure: We'll meet once or twice a week and discuss what you've coded, what it means, and where it makes sense to go next. Everyone's journey will be individual, so meetings will likely be one-on-one.

Expected Input: Coding! Probably about 3 hours/week total, including meetings.

Expected Output: Pretty charts and an ocean of unprovable conjectures.

Difficulty: 🌶🌶🌶–🌶🌶🌶🌶

Prerequisites: Computer programming. You should be able to do math computations in some programming language (enough to answer questions like "How many twin primes are there that are less than 10^8?" or "What's the average number of squarefree numbers in an interval $(n, n+1000]$ for $1 \le n \le 10^8$?"), and be comfortable making charts.

Yuval's Projects

Linear B (Yuval)

Description: The decipherment of Linear B is one of the crowning achievements of archeology and historical linguistics of the 20th century. It's also a great topic to learn for people who like math, because a lot of the key insights came down to "think really rigorously about how things should work, rather than just making stuff up and hoping for the best." Additionally, there is a very good and gripping account of the story, *The Decipherment of Linear B* by John Chadwick.

In this project, you will read through Chadwick's book, and we'll meet weekly to discuss the reading. Additionally, we'll read some other sources to fill in the holes in Chadwick; most notably, we'll try to read at least one of the original papers by Alice Kober, who gets short shrift in Chadwick and whose work deserves to be more seriously understood.

Structure: Weekly meetings, plus a fair amount of reading

Expected Input: 3 hours per week of reading, plus 1 hour for the meetings

Expected Output: By default none, but we can discuss options if you're excited! You can make a poster for the project fair, or we could even try making clay tablets and inscribing them with some Linear B writing of our own.

Difficulty: 🌶

Prerequisites: None

Number squares (Yuval)

Description: Start with four numbers on the vertices of a square. At the midpoint of an edge, write the absolute difference of the values on its endpoints (i.e. if two adjacent corners are a, b, then write $|a - b|$ on the midpoint). This gives a new square with four numbers on its corners, so we can repeat. What happens to this process? For example, starting with the numbers $(1, 5, 3, 6)$, we have the process

$$(1, 5, 3, 6) \mapsto (4, 2, 3, 5) \mapsto (2, 1, 2, 1) \mapsto (1, 1, 1, 1) \mapsto (0, 0, 0, 0),$$

at which point all further iterations remain at $(0, 0, 0, 0)$.

There are lots of other related questions one can ask. For example, do things change if the starting values are integers, rational numbers, real numbers, or complex numbers? What happens if we replace a square by another polygon? Can we make any sense of this in higher dimensions?

Structure: You'll work these problems, and meet with me to discuss your progress.

Expected Input: 2–4 hours per week

Expected Output: A poster for the project fair; it might include theorems proved, computer simulations simulated, or conjectures conjectured.

Difficulty: 🌶🌶

Prerequisites: None

Zoe's Projects

Picture book mathematics (Zoe)

Description: For some people thinking about math means that they are constructing some type of picture in their head or sketching concepts out on paper. In this project we will take this to the extreme and will be attempting to translate the explanations of our favorite mathematical objects or favorite proofs almost exclusively through pictures with the end goal of making some "picture book" that covers technical math concepts.

Structure: The first meeting will likely be deciding the structure of the book, then there will be weekly meetings where people discuss and give feedback on whether each figure is communicating all the things we want it to and stuff like that. There will probably also be weekly optional coworking sessions that can also act more as "artistic support" for anyone who might want to learn some extra TikZ or drawing skills. Lastly there will be a final meeting (or two meetings) where the goal will be to streamline the end product.

Expected Input: Drawing figures and giving input on other peoples figures. Probably a time commitment of 2–4 hours

Expected Output: A picture book that walks people through the selected proofs

Difficulty: 🌶

Prerequisites: None

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.