Welcome!

Introduction to Mathematical Thinking

January 29th, 2019

Suraj Rampure

Why does this course exist?

This is the second semester of the course, but the course has been in progress since late 2017!

1. CS 70 is hard! We're here to help. By exposing you to difficult concepts in advance, you'll have time to think about them without the stress of declaring.
2. Math, especially discrete math is fun! We want you to feel the same way.
    - In an ideal world, non-CS/math students could take this course just for fun

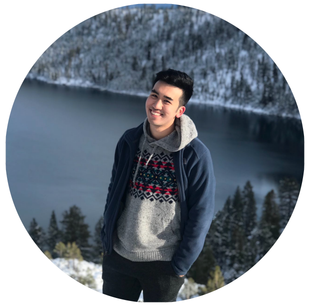
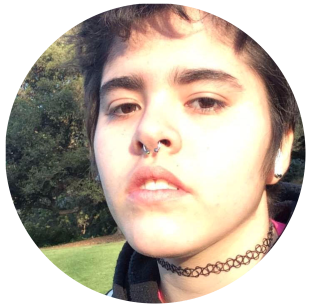
This is a 2-unit DeCal.

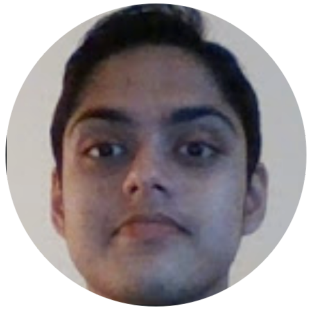
- With that said, the best way to learn math is by doing math. You **will** have to work for a P in this class, but it's for your own good.

Meet the Teaching Staff

Instructor: Suraj Rampure

TAs: Adel, Alexia, Divya, Jai, Sagnik

Alexia, Divya, Jai, and Sagnik took this course last semester!

Syllabus

Of course, this is still a new class, so the syllabus is subject to change. An up-to-date version will remain on our course website.

Topic	# of Lectures
Set Theory and Functions	3
Propositional Logic	2
Proofs (basic proofs, induction, applications)	5
Number Theory and Modular Arithmetic	3
Counting Techniques	4
Combinatorics with Polynomials	4
Polynomial Interpolation, Factor and Remainder Theorems	2

Logistics

Class will be held on Tuesdays and Thursdays from 3:30-5:00PM in LeConte 2.
Attendance is mandatory: let us know in advance if you can't make it.

- Office Hours: Monday 3-4PM, Wednesday 7-8PM, Thursday 2-3PM, all in Soda 341B
- We will use Piazza, you will receive invites today
- Course website is at **http://imt-decal.org**, where all content is posted (including videos, textbook readings)
- Textbook is at **http://book.imt-decal.org**
- **imt-decal@berkeley.edu** is your main point of contact

Grading

Course is out of 100 points, need 75 to pass.

- **5 in-class quizzes** (12 points each, totalling 60 points)
- **Weekly homework** graded on effort (totalling 40 points)

Doing homework forces you to engage with the material each week. It's graded on effort, so you don't need to spend hours on end working on it, but we do recommend you come to office hours with questions. Homeworks will follow a Friday-Friday schedule, and will be collected on Gradescope.

This semester, we've decided to have regular quizzes as opposed to a midterm and final (last semester). We feel that quizzes work better both from a pedagogical and logistical standpoint.

Quiz Dates

1. Tuesday, February 12 OR Thursday, February 14
    - Feb. 12 is one day before the drop deadline, but the day after CS 61A MT1.
    - Feb. 14 is one day after the drop deadline, but a few days after CS 61A MT1.
    - We need your opinion! Survey will be on the attendance form.
2. Thursday, February 28
3. Thursday, March 14
4. Tuesday, April 9
5. Tuesday, April 30

Quizzes will be 30 minutes long, and will be held at the start of class (3:40-4:10).

Attendance

Fill out the form at **http://tinyurl.com/imtfirstday**. You will be dropped from the class if you don't fill this out!

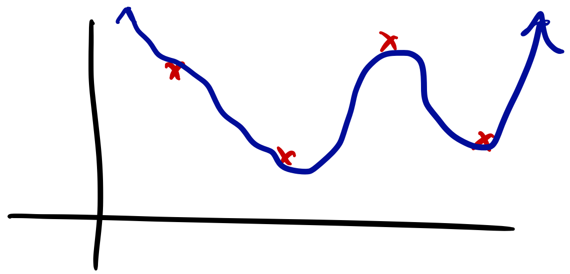
Attendance is especially important this week, as we'll be emailing out enrollment codes by tomorrow. **Remember to enroll in the units before the add/drop deadline (February 13).** If you no longer intend on being in the course, please email us ASAP.

For the rest of today's class, we'll look at a few cool applications of some of the principles we'll see in this class.

You won't be tested on any of this content. It serves as a preview of what's to come in the course. Sit back and enjoy!

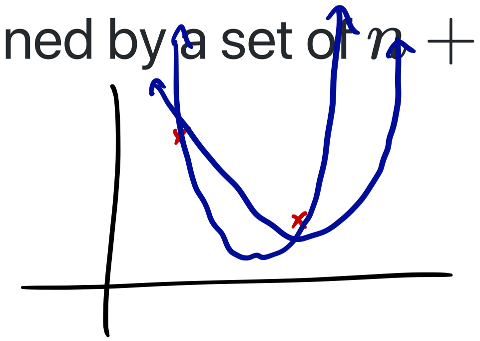
Demo – Polynomial Guessing

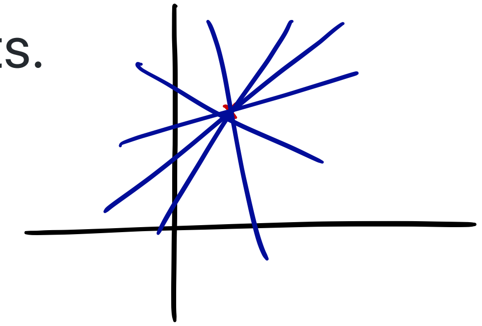
Background Material – Polynomials



A polynomial is a function of the form $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0)$.

A polynomial of degree n can be uniquely determined by a set of $n + 1$ points.





For example:

- The degree-1 polynomial $f(x) = 3x - 4$ can be uniquely determined by the set of points $\{(0, -4), (3, 5)\}$.
  - However, there are infinitely many points passing through $(0, -4)$. $f(x)$ is one of them, but not the only one.
- The degree-3 polynomial $g(x) = 2x^3 - 12x + 21$ can be uniquely determined by the set of points $\{(1, 11), (-1, 31), (2, 13), (-2, 29)\}$.
  - However, there are infinitely many points passing through the set of points $\{(1, 11), (-1, 31), (-2, 29)\}$. $g(x)$ is one of them, but not the only one.

We will formally define a set as well as the above properties later in the course. Take the leap of faith for now.

I'm a magician!

Let's play a game.

Think of any polynomial with **non-negative, integer coefficients**, preferably of degree 2 or more.

- For example, $p(x) = 3x^4 + 4x^2 + x + 5$.

I'm going to guess your polynomial by asking for **just two points**, regardless of the degree of your polynomial.

Background Material – Numerical Bases

$6^4 < 1344 < 6^5$

Don't worry too much if this seems out of reach – we will revisit bases when we talk about number theory and modular arithmetic.

In school, you were taught that we can write any integer as a sum of powers of 10. For example,

$$1344 = 1 \cdot 10^3 + 3 \cdot 10^2 + 4 \cdot 10^1 + 4 \cdot 10^0$$

This is because our standard convention is to use **base 10**. However, we can write numbers in any base we want! If we wanted to write 1344 in **base 6**, i.e. 1344_6, this job boils down to finding coefficients such that

$$1344 = a \cdot 6^4 + b \cdot 6^3 + c \cdot 6^2 + d \cdot 6^1 + e \cdot 6^0$$

Since 1344 is between 6^4 and 6^5, we know that the largest power of 6 we will include is 4. Note that the largest possible "digit" in base 10 is 9 – in general, the largest possible digit in base b is $b - 1$.

There is a standard process we can use to find these coefficients. We can repeatedly divide by b.

Rewriting Integers in Base n

% → remainder when divided by

Let's attempt to rewrite 1344 in base 6.

largest poss digit → 5

$$1344 = a \cdot 6^4 + b \cdot 6^3 + c \cdot 6^2 + d \cdot 6^1 + e \cdot 6^0$$

$$1344_{10} = 10120_6$$

1344 // 6 = 224

224 // 6 = 37

$f(x) = 1x^3 + 12x + 1$ $f(1) = 1 + 12 + 1 = 14$

$f(1) = 5$

$f(6) = ?$

Back to the problem... How did I do it?

I asked for two very specific points – $f(1)$ and $f(f(1)+1)$.

$b = f(1)$ is the sum of the coefficients of f. I know that the largest possible coefficient is b, since they're all positive. If the largest coefficient is b, I know that the coefficients can represent a number in base $b+1$ or greater.

- To illustrate, "133" can represent a number in base 4 or higher, but not in base 2, 3

Then,

$$f(b+1) = a_n(b+1)^n + a_{n-1}(b+1)^{n-1} + a_{n-2}(b+1)^{n-2} + ... + a_1(b+1)^1 + a_0$$

Finding $a_n, a_{n-1}, ..., a_1, a_0$ is the same problem as writing $f(b+1)$ in base $b+1$. We know this is possible, since none of the coefficients are larger than b!

Let's work out an example.

$(1023)_7 = 1\cdot 7^3 + 0\cdot 7^2 + 2\cdot 7^1 + 3\cdot 7^0$

$f(x) = x^3 + 2x + 3$

Suppose my function is $f(x) = x^3 + 2x + 3$.

I ask for $f(1)$ and get 6.
I ask for $f(7)$ and get 360.

rewrite 360 in base 7

$360 // 7 = 51$

Hexadecimal

0 → 0
⋮ → ⋮
9 → 9
A → 10
B → 11
⋮
F → 15
⋮

1023

$f(x) = x^n$ $f(1) = 1$ $f(1) = 1$ $f(2) = 512 \rightarrow f(x) = x^9$

- I also instead could have asked for $f(2)$ and $f(f(2)+1)$, or $f(3)$ and $f(f(3)+1)$ – all we needed was that the first query was at least as large as the largest coefficient.
- Sometimes, in this trick, we're asked for $f(1)$ and $f(f(1))$, but there are a few edge cases to handle.
  - If $f(x) = cx^n$, this will not work. This is because the largest coefficient is equal to the sum of coefficients. $f(1) = c$, and $f(c) = c^{n+1}$. This result would lead you to guess that the polynomial is $f_{guess}(x) = x^{n+1}$, though you can verify pretty quickly that this doesn't hold for $f_{guess}(1)$ and conclude that $f(x) = cx^n$.
  - Also, this doesn't work for $f(x) = x^n$, as $f(1) = 1$ and $f(f(1)) = 1$ both query for the same point, $(1, 1)$. Instead, if $f(1) = 1$, since we know all coefficients in this trick are non-negative, our polynomial will not be of the form $p(x) = 2x^3 - 1$ (for example), so we can conclude our polynomial is of the form $f(x) = x^n$ and simply ask for $f(1)$ and $f(2)$.

Let's write a function that implements this in code!

Does this violate the properties of polynomials?

Does this violate the properties of polynomials?

No!

- The property that a degree n polynomial requires $n + 1$ points to be identified refers to any set of $n + 1$ *arbitrary* points, i.e.some set of points that we are given
- Our trick relies on the fact that our second "query" depends on the first one – we aren't using an arbitrary set of points

Note: Instead of writing our own code, we could have just used https://www.rapidtables.com/convert/number/base-converter.html – but that's not as fun!

Next Time

- We will begin Sets and Functions – feel free to read 1.1 and 1.2 in the textbook (but not necessary – we will cover it all)
- This week:
  - Permission codes will be sent out
  - Piazza and Gradescope will be set up
  - HW 1 will be released on Friday!