Math 127, Mon Feb 08

- Use a laptop or desktop with a large screen so you can read these words clearly.
- In general, please turn off your camera and mute yourself.
- Exception: When we do groupwork, please turn both your camera and mic on. (Groupwork will not be recorded.)
- Please always have the chat window open to ask questions.
- Reading for today: 3.1.
- Reading for Wed: 3.2–3.3.
- PS01 due today; PS02 outline due Wed, full version due Mon Feb 15.
- Next problem session Fri Feb 12, 10:00–noon on Zoom.
- **Exam 1** in 2 weeks from today.

How to solve every (substantial) problem in algebra and combinatorics:
(Problem = you don't know the method for solving beforehand)

Try examples until your ears bleed! And then look for a pattern.

2.3.4: (a) Find a, d s.t.

$$a = dq + r \qquad |r| \le \frac{|d|}{2}$$

has > 1 ans.

Take d = 5, and try a = 7, 8, 9, 10, 11....
Take d = 6, and try a = 10, 11, 12, 13...

In each case, look for more than one possible q. (How do we get q in the first place?)

Recap of complexity

The **complexity** of an algorithm is the (estimated) time or space that an algorithm needs to finish, given an input of size n.

This is often expressed in **big O notation**. That is, we say that a given algorithm finishes in $O(f(n))$ time, meaning that there exists some constant C such that the number of steps the algorithm takes is $\leq Cf(n)$ for all $n \in \mathbf{N}$.

Last:
Naive gcd is O(n).
PS01 gcd is O(sqrt(n)).

The Euclidean Algorithm is exponentially faster

Theorem

Let a, b, and n be nonzero integers with $|a| \geq |b|$ *and* $|b| \leq n$. *Using the Signed Euclidean Algorithm to compute* $\gcd(a, b)$ *finishes in* $O(\log n)$ *time, or more precisely, requires* $O(\log n)$ *division-with-remainder steps to finish.*

Idea of proof: Look back at the Signed Euclidean Algorithm.

$r_0 = b \leq n$

Point is: Remainder cut in half with each step.

k steps

$$|r_1| \leq \frac{|r_0|}{2} \leq \frac{n}{2}$$

$$|r_2| \leq \frac{|r_1|}{2} \leq \frac{n}{4}$$

$$|r_3| \leq \frac{n}{8}$$

$$\vdots$$

$$1 \leq |r_{k-1}| \leq \frac{n}{2^{k-1}}$$

$$n \geq 2^{k-1}$$

$$k \leq \underbrace{\log_2 n + 1}_{O(\log n)}$$

Note: In big-O notation, all logs are the same, since log_a(x) is just log_b(x) times a postive constant. (Change of base formula)

Ex $\gcd(132, 53)$

$132 = 2(53) + 26 \qquad 26 \leq \frac{53}{2}$

$53 = 2(26) + 1 \qquad 1 \leq \frac{26}{2}$

If you want to see a very physical example of exponential speedup in sorting (n log n vs. n^2), try sorting a deck of cards using merge sort.

(Look up what a merge sort is online, or come to office hours.)

Miryam's example: Searching a sorted list/tree vs. searching an unsorted list.

Another example of a complexity estimate

The traditional Christmas carol "The 12 Days of Christmas" has the following structure: On day 1, the singer gets one gift of type 1 (a partridge in a pear tree) from their true love; on day 2, the singer gets two gifts of type 2 and one gift of type 1 (two turtledoves and a partridge in a pear tree); and so on. Suppose this song can be extended to any arbitrary number of days.

- Give a big-O estimate of the *number* of gifts the singer receives on day *n*.
- Give a big-O estimate of the *total number* of gifts the singer receives over the entire song, going from day 1 through day *n*.

day n: $1+2+3+\cdots+n$ gifts

Ans 1 $\text{sum} = \frac{n(n+1)}{2} = \cancel{\frac{n^2+\cancel{n}}{2}} = O(n^2)$

Ans 2 $\quad \underbrace{1+2+3+\overbrace{\cdots+n}^{\text{half}}}$ $\geq \frac{n}{2}$

Each $\leq n$, n terms

So $\leq n^2 = O(n^2)$

NB! Sum $\geq \left(\frac{n}{2}\right)^2 = \frac{n^2}{4}$

Ans Each day $\leq O(n^2)$ gifts

n days

$\boxed{\text{So } O(n^3).}$

Note: Can show $\Theta(n^3)$.

And now for something completely different

Ch.3

Recall that the **ring** in which we work is the set of "numbers" we're allowed to use in computations, proofs, etc.
How do we change the ring we work in? I.e., how do we define a **new** ring? To define a ring R:

- *Choose a set:* First, choose a set R of objects that will be the "numbers" of your ring.
- *Define addition:* Next, define how to add two elements of R.
- *Define multiplication:* Finally, define how to multiply two elements of R.

If you can do this in a way that allows you to use the rules of high school algebra consistently, then you get a ring where we can do reasonable work.

Reduction mod m

We'll be defining a new ring $\mathbf{Z}/(m)$, but first, a pre-definition:

Definition

Let m be a positive integer. For any integer k, k **reduced (mod m)** is the remainder you get when you divide k by m. I.e., if

$$k = qm + r \qquad \text{with } 0 \leq r < m,$$

then k reduced (mod m) is r.

Example: $m = 11$

63 reduced mod 11: 63 = 5(11) + 8, so 63 reduced mod 11 is 8.
23 reduced mod 11: 23 = 2(11) + 1, so 1.

-17 reduced mod 11: -17 = (-2)11 + 5, so 5.

Alt method for negative integers: Keep adding 11 until you get a nonneg number. Example: What is -432 reduced mod 11?
Well, -432 + 440 = 8. 440 is a multiple of 11, and 0 <= 8 < 11, so must be 8.

Write: $-432 = 8 \pmod{11}$

<u>Alt</u>: $432 = 39(11) + 3$

$$432 = 3 \pmod{11}$$

$$-432 = -3 = 8 \pmod{11}$$

$+11$

$\mathbf{Z}/(m)$, the integers mod m

unusual not'n

Let m be a positive integer. We define the ring $\mathbf{Z}/(m)$, or the **integers (mod m)**, as follows.

- The underlying set of $\mathbf{Z}/(m)$ is $\{0, \ldots, m-1\}$.
- For $a, b \in \mathbf{Z}/(m)$, we define $a + b$ to be the ordinary integer sum of a and b, reduced mod m.
- Similarly, for $a, b \in \mathbf{Z}/(m)$, we define the product ab to be the ordinary integer product of a and b, reduced mod m.

When we work in $\mathbf{Z}/(m)$, we refer to m as the **modulus** of our ring.

"work mod m"

Example: Addition and multiplication tables in $\mathbf{Z}/(5)$

$\mathbb{Z}/(5) = \{0, 1, 2, 3, 4\}$

mod 5

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3					
4					

·	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

sudoku!

How to be more flexible

For $m = 3$ (as an example), since $-1 = 2 \pmod 3$, it sometimes helps to write the elements of $\mathbf{Z}/(3)$ as $\{-1, 0, 1\}$ instead of $\{0, 1, 2\}$.
More generally:

> **Congruent Substitution Principle:** *If we are working in the ring $\mathbf{Z}/(m)$, we can always replace any integer a with any integer b congruent to $a \pmod m$.*

Even more generally:

> **The $m = 0$ Principle:** *Arithmetic in $\mathbf{Z}/(m)$ is like regular arithmetic, except that we declare that $m = 0$, and accept all of the relations that follow as a consequence (such as the Congruent Substitution Principle).*

Example: Fractions in $\mathbf{Z}/(7)$

In $\mathbf{Z}/(7) = \{0, 1, 2, 3, 4, 5, 6\}$, what is the reciprocal of each element?

$2 \cdot 4 = 1 \quad (\text{in } \mathbb{Z}/(7))$

$\frac{1}{2} = 4 \quad \text{in } \mathbb{Z}/(7)$

Experiment 1: Primitive elements

Defn: To say that $a \in \mathbf{Z}/(m)$ is **primitive** means that every nonzero element of $\mathbf{Z}/(m)$ is a power of 2.

- Is 2 primitive in $\mathbf{Z}/(5)$?
- Is 2 primitive in $\mathbf{Z}/(7)$?
- Is 2 primitive in $\mathbf{Z}/(11)$?

Experiment 2: Quadratic residues

Defn: To say that $a \in \mathbf{Z}/(m)$ is a **quadratic residue** means that $a \neq 0$ and a is a square in $\mathbf{Z}/(m)$, i.e., $x^2 = a$ has a solution in $\mathbf{Z}/(m)$.
Here we use a weird method for solving $x^2 = a$: Because there are only finitely many values of $x \in \mathbf{Z}/(m)$, we can just try all possible x.

- Quadratic residues in $\mathbf{Z}/(5)$:
- Quadratic residues in $\mathbf{Z}/(7)$:

The point of the last few problems in PS02: Experiment!

Try a bunch of examples and see if you find any patterns!

(And yes, the other point is for you to get better at computation in $\mathbf{Z}/(m)$ through practice — but you might as well do something interesting in the process.)