Recap:

1. gcd(n,m): largest number that divides n and m.

2. Useful identity: $\forall\, n, m$, $\exists\, s, t$ s.t.
$$\gcd(n,m) = s\cdot n + t\cdot m$$

Exercise (9.12 [LLM]):

2 player game. Player 1 chooses two natural numbers n, m, Player 2 chooses who plays first. Initially, n, m are written on the blackboard. Each player, during their turn, must write a NEW number >0 that is the difference of some two numbers on the board. The person who is not able to produce such a number loses.

eg

P_1: 9, 15	11, 18 : P1
↓	↓
P2 : 6, 9, 15	7, 11, 18 : P2
↓	↓
P1 : 3, 6, 9, 15	4, 7, 11, 18 : P1
↓	↓
P2 : 3, 6, 9, 12, 15	4, 7, 11, 14, 18 : P2
	↓
P1 loses	3, 4, 7, 11, 14, 18 : P1

<u>Lemma 8.1</u> : Player 2 can always win.

Initially, there are 2 numbers.

Suppose $m \geq n$. Eventually, we will have

$z = (m / \gcd(n,m))$ numbers, containing all multiples of $\gcd(n,m)$ upto m (and nothing else). If z is odd, P1 plays first, else P2 plays first.

??

<u>Proof of Lemma 8.1</u> : We will prove this using the following claims.

<u>Claim 8.1</u> : At any stage, $\gcd(n,m)$ divides all numbers.

Proof by induction.

$P(k)$: $\forall\ n, m$, for any strategy, the set of numbers written after k^{th} step are divisible by $\gcd(n,m)$.

Base case $k=1$: $n, m, m-n$ are the three numbers in step 1. $\gcd(n,m)$ divides them all.

Suppose $P(k-1)$ holds.

Take any n, m, and take any strategy. Let $a_1, a_2, \dots, a_{k+1}$ be the numbers before k^{th} step. In k^{th} step, suppose we compute $a_i - a_j$ for some i, j. Then $\gcd(n, m)$ divides all a_j, $j \le k+1$ (since we assumed $P(k-1)$).
Next, note that $\gcd(n, m)$ also divides the new number $a_i - a_j$. Hence $P(k)$ holds.

Hence, using induction, $\forall k$, $P(k)$ holds. $\blacksquare$

<u>Observation</u>: The game terminates in at most $\max(n, m)$ steps.

<u>Proof</u>: Every step adds one new number to the sequence. Every number is at least 1, and at most $\max(n, m)$. Hence, game terminates in at most $\max(n, m)$ steps. $\blacksquare$

Since the game terminates in finite steps, the "final sequence" is well defined.

Claim 8.2: Suppose the final seq of numbers is $a_1 < a_2 < \cdots < a_t$. Then $a_j = j \cdot a_1$ for all j.

Proof: Proof by contradiction.

Suppose $\exists j$ s.t. $a_j \neq j \cdot a_1$.

Consider the smallest such j. Note that $a_1 = 1 \cdot a_1$, therefore $j \geq 2$.

$a_{j-1} = (j-1) a_1$ but $a_j \neq j \cdot a_1$.

Two possibilities.

$a_j > j \cdot a_1$. Then $(a_1 \cdots a_t)$ is not the final sequence. We can have next seq.
$(a_1, a_2, \cdots, a_{j-1}, a_j - a_1, a_j, \cdots, a_t)$

Note that this is a new sequence because $a_{j-1} < a_j - a_1 < a_j$

$a_j < j \cdot a_1$. Consider $a_0 = a_j - a_{j-1}$.
$a_0 < a_1$, and therefore $(a_1 \cdots a_t)$ is not the final sequence, as
$(a_1 \cdots a_t) \longrightarrow (a_0, a_1, \cdots, a_t)$.

Hence, contradiction. ∎

a_1 div. all num. in final list, which includes n and m

From Claim 8.2, we get that a_1 divides both n and m, and therefore $a_1 \leq \gcd(n, m)$. From Claim 8.1, we get that $\gcd(n, m)$ divides a_1, and therefore $\gcd(n, m) \leq a_1$. Therefore, $\gcd(n, m) = a_1$, and the final sequence consists of all multiples of a_1 that are at most m.

∎

Some open-ended questions to conclude our discussion on gcd:

<u>Qn 1</u>: Let $\gcd_n(a_1, a_2, \ldots, a_n) = \left\{ \begin{array}{l} \text{largest } d \text{ that} \\ \text{divides all } a_i \end{array} \right\}$

$\gcd_n(a_1, a_2, \ldots, a_n)$ can be computed efficiently using Euclid GCD algorithm.

Consider $\gcd_{n/2}(a_1, a_2, \ldots, a_n) = \left\{ \begin{array}{l} \text{largest } d \text{ that} \\ \text{divides at least} \\ n/2 \text{ of the } a_i\text{s} \end{array} \right\}$

(i) Can $\gcd_{n/2}(a_1, a_2, \ldots, a_n)$ be computed efficiently?

(ii) can $\gcd_{n/2}(a_1, a_2, \dots, a_n)$ be computed efficiently, if you are given the prime factorization of the a_i s?

(iii) Suppose $\gcd_{n/2}(a_1, a_2, \dots, a_n)$ can be computed efficiently, using algorithm A. Can we use A to find prime factorization of any natural number?

Q2: Approximate GCD

Suppose you are given n natural numbers $x_1 \dots x_n$ s.t. $x_i = p \cdot q_i$, and suppose you are also given that $\gcd_n(q_1 \dots q_n) = 1$.

Your job is to find p. Easy?

Now consider the following problem:

Given: $x_1, x_2, \dots, x_n$ where

$x_i = p \cdot q_i + r_i$

$n \sim 10000$.

You are also given that $p \sim 2^{1000}$

all $q_i < 2^{1000}$, all $-2^{100} < r_i < 2^{100}$.

Goal: find p.

This problem is believed to be computationally hard, and is used to build public key encryption schemes (we may see this in Tutorial 3)

MODULAR ARITHMETIC

Let $n \in \mathbb{N}$. The set of all possible remainders when dividing by n, denoted by $\mathbb{Z}_n = \{0, 1, 2, \ldots, n-1\}$, can have interesting properties, depending on the structure of n.

We will study $\mathbb{Z}_n$ when n is prime, and $\mathbb{Z}_n$ when n is product of two primes. Both have immense practical applications, AND ALSO RICH MATH. STRUCTURE.

Puzzle 1 : Dealing with top secrets

I have a 100 bit secret s. I want to 'distribute' this share among the class students s.t.

- if all students are present, then should be possible to recover s.

- if even one student missing, then the remaining students, even together, should not learn even 1 bit of information about s.

Suppose there are n students. Pick $n-1$ 100-bit strings, uniformly at random — $r_1, r_2, \dots, r_{n-1}$.

Give r_i to i^{th} student.

Give $s \oplus \left(\bigoplus_{i=1}^{n-1} r_i \right)$ to n^{th} student.

Suppose, I want to distribute the secret s among n people s.t.

- any subset of t people should be able to reconstruct the secret.
- any subset with less than t people should learn nothing about s.

Puzzle 2: ERROR CORRECTING CODES

Alice Bob

Alice wants to send a message to Bob over a lossy channel. The channel "corrupts" k of the packets. How should Alice "encode" her message?

Error-correcting codes are widely used in practice
eg QR codes

Most natural idea:

$m \xrightarrow{\text{Encode}}$ $2k+1$ copies of m.

$\downarrow$ sent over lossy channel

Can we do better?

$\downarrow$ Decode by taking majority

m