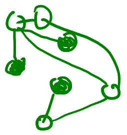
Lec. 24: P vs. NP

Date: 01/02/23

There are some extremely hard problems with probably no efficient solution **but** no one has been able to prove that so far!

Problem

Optimization Problem | Decision Problem



Ind. Set

Find largest ind. set in graph G.

Does there exist an ind. set of size $\geq k$?

Yes No

25

5 15

Solution to opt. version of ind. set $\Longrightarrow$ sol. to dec. version for all k

Solution to $O(\log n)$ dec. versions of ind. set $\Longrightarrow$ sol. to optim. version of ind. set.

There are many "hard" problems.

Exponential Time:
- TSP
- Ind. Set
- 0/1 Knapsack
- Subset Sum
- Graph Coloring

Easy Problems

Polynomial:

Linear Search	$O(n)$
Binary Search	$O(\log n)$
Quicksort	$O(n^2)$
Mergesort	$O(n \log n)$
Matrix Mult.	$O(n^3)$
n-digit mult.	$O(n^{1.59})$
Stable Matching	$O(n^2)$

- Let P be the set of problems with polynomial time solutions.
- Let NP ———— exponential ———— **BUT** polynomial time verifications of solutions

For example,

- Ind. Set is $O(2^n)$ but given a solution, we can verify in polynomial time that given sol. is ind. or not.
- TSP is $O(2^n)$ but given a solution, we can verify in poly. time that it goes from 1, to every other city once and only once, and ends at 1.

Non-deterministic Polynomial

NP-Complete

Bipart. Matching → IS
Int. Sched. → IS

- There are some problems in NP such that all problem in NP can be reduced in polynomial # steps to these problems.
- If any problem in NP-Complete has a polyn. time Solution, then all problems in NP will have poly. time solutions, since they are reducible to NP-Complete in poly. time.
    - As a result, set P will equal set NP

    - Proving P = NP or even P ≠ NP is a million dollar problem, literally.

Millenium Prize — 10 problems → P = NP

① 0/1 Knapsack
② SATisfiability $O(2^n)$
← reduction
reducibility