Types of Problems

1 million dollars!

Easy

(Polynomial time)

- Search
- Sort
- Multiplication
- Closest Points
- Greedy Scheduling
- MWIS on a line
- Matrix Mult.

? ?? ?

Puzzles

Crossword

Sudoku

Delivery rt ≤ 100 miles

Protein Folding

$\boxed{\text{Factor larger numbers}}$

Quantum

Trade sequence ≥ $1 Million

MWIS ≥ 1000

- In a puzzle, the solution can be revealed and you can easily check the solution
- Lots of real world important problems

Hard

Chess

What is next best move?

QRS

How to even check if correct? Need to check all possible responses, responses to responses...

Can mathematically characterize Easy / Puzzle

P and NP

P (Polynomial Time)

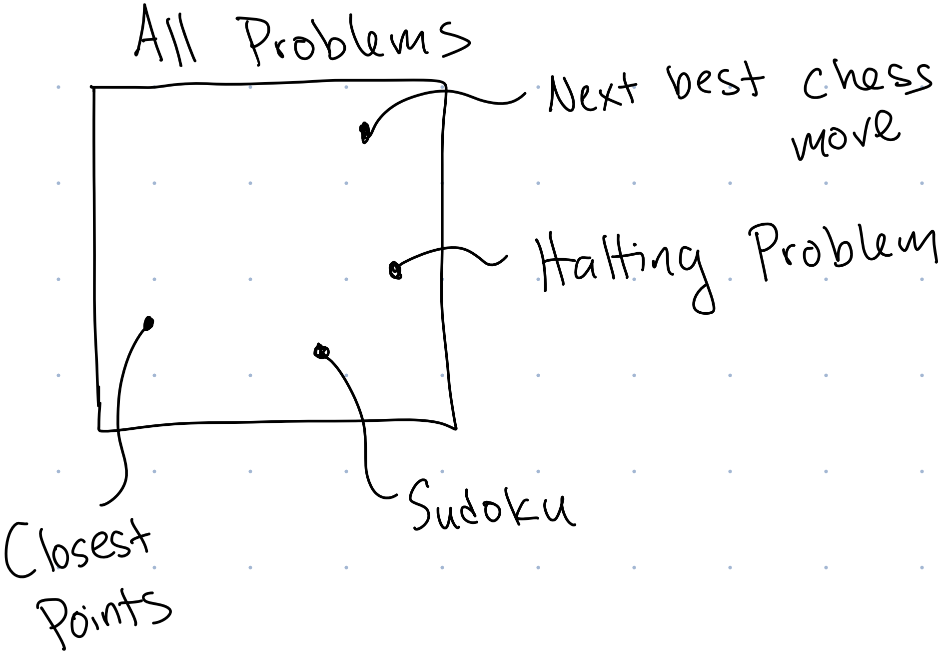
Informal: A problem is in P if it can be solved in polynomial time

NP (Non-deterministic Polynomial Time)

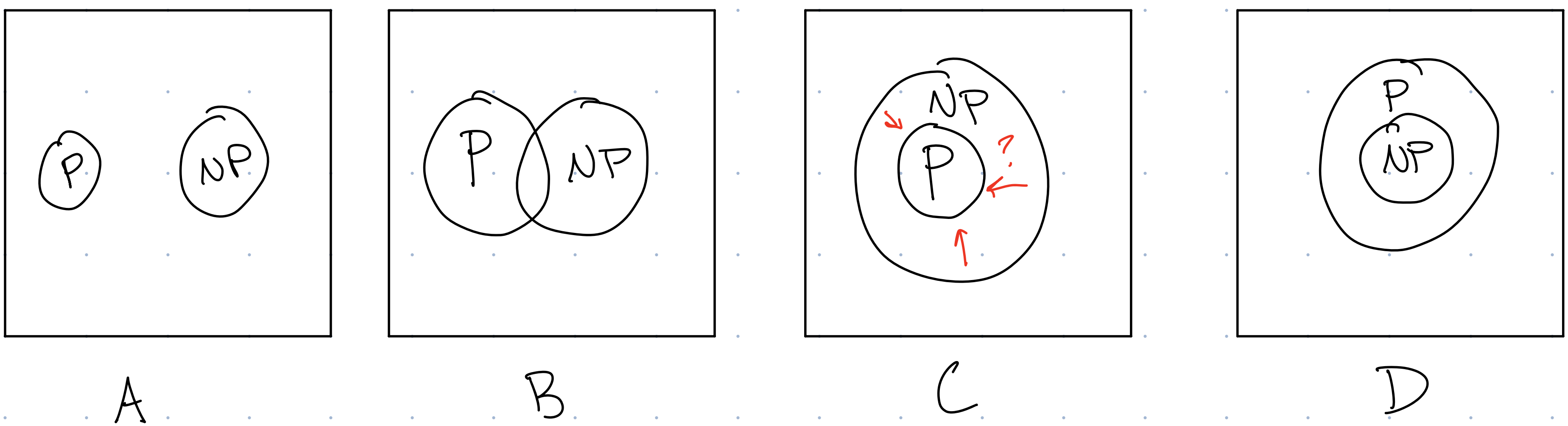
Informal: A problem is in NP if solution can be checked in polynomial time

Polynomial Time ??

- $O(n^c)$ for a constant c, where $n = \#$ of bits used to describe input.



Which picture is correct?



C: Every problem in P is also in NP because if you can find a solution you must have been able to check its correctness

Formal(ish) Definition of NP

"Decision Problem"

"verifier" checks if solution is correct

$O(poly(|x|))$

"witness" the solution

A problem is in NP if
- Yes-No problem
- There is a polytime algorithm M s.t.
  - If x is a YES instance, $\exists\, y$ s.t. $M(x,y)=1$
  - If x is a NO instance, $\forall\, y$, $M(x,y)=0$]

Verifier can't be fooled

example of NP problem:

<u>3SAT</u>: x is a YES instance if it describes Boolean formula that is an AND of ORs where each clause has at most 3 literals and there is an assignment of variables that makes x true

Instance:

$$x = (z_1 \vee z_2 \vee \neg z_3) \overset{\text{AND}}{\wedge} (\neg z_1 \vee \neg z_3 \vee z_4) \wedge (z_2 \vee z_4) \wedge \ldots$$

OR   literals   variable

Expected Witness:

$$y = (z_1 = T,\ z_2 = F,\ z_3 = F \ldots)$$

Proof that $3SAT \in NP$

- Let $M(x,y)$ be the algorithm that
  1. Checks that x is the AND of a series of OR clauses of where each clause has at most 3 literals
  2. Checks that y is an assignment of T, F to the n variables
  3. Checks that with the assignment every clause is true

  And outputs 1 if all checks pass, else outputs 0.
- $M(x,y)$ runs in $O(|x|^2)$ time:
  1. Read through formula in $O(|x|)$ time (x is formula) + check form
  2. Read assignment in $O(|x|)$ time (# variables $\leq$ size of formula)
  3. $\left.\text{For } \underbrace{\text{each clause}}_{O(|x|)}\text{, check if } \underbrace{\text{assignment}}_{O(|x|)} \underbrace{\text{makes it true}}_{O(1)}\right\} O(|x|^2)$
- If x is a YES instance, $\exists\, y$ s.t. $M(x,y) = 1$, by choosing y to be satisfying assigment
- If x is a NO instance, either x is not a valid formula or no satisfying assignment and $M(x,y)$ will never pass all checks for any y.

Group Work

Hamiltonian Path Problem: X is a YES instance iff X describes adjacency matrix of a graph G with vertices s, t s.t. there is a path from s to t that goes through each vertex exactly once.

$X =$

	s	u	v	t
s	0	1	0	0
u	1	0	1	1
v	0	1	0	1
t	0	1	1	0

$\Uparrow$ YES instance

$X =$

	s	u	v	t
s	0	1	1	1
u	1	0	0	0
v	1	0	0	1
t	1	0	1	0

$\Uparrow$ NO instance

Show:

Prove: Hamiltonian Path $\in NP$

- Describe $M(x, y)$
- Analyze runtime of M in terms of $|x|$
- If yes, $\exists\, y : M(x, y) = 1$
- If no, $\forall\, y : M(x, y) = 0$

$\boxed{\text{Is Knapsack in } P?\ NP?}$

Group Work

Hamiltonian Path Problem: Given an adjacency matrix for a graph $G=(V,E)$ and $s,t \in V$, is there a path from s to t that goes through each vertex once.

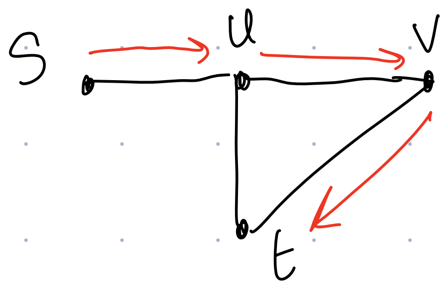
$$X = \begin{array}{c|cccc} & s & u & v & t \\ \hline s & 0 & 1 & 0 & 0 \\ u & 1 & 0 & 1 & 1 \\ v & 0 & 1 & 0 & 1 \\ t & 0 & 1 & 1 & 0 \end{array}$$

$\Uparrow$ YES instance

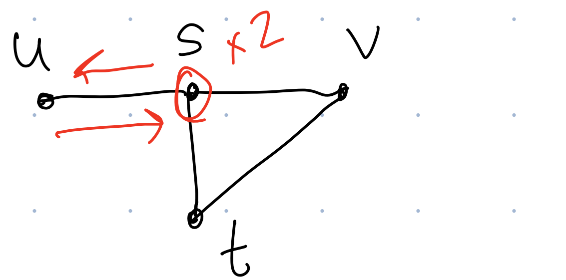
$$X = \begin{array}{c|cccc} & s & u & v & t \\ \hline s & 0 & 1 & 1 & 1 \\ u & 1 & 0 & 0 & 0 \\ v & 1 & 0 & 0 & 1 \\ t & 1 & 0 & 1 & 0 \end{array}$$

$\Uparrow$ No instance

Show:



$y = (s, u, v, t)$



Ham Path $\in$ NP

- $M(x,y)$:

① Checks x is adjacency matrix with vertices s, t, n total vertices

② y is a list of n vertices starting with s, ending t.

③ y has no repeated vertices

④ For each consecutive pair $(u,v) \in y$, (u,v) is edge in G.

If passes all checks, return 1, else, 0

- Runtime:

① $O(n^2)$: Size of adjacency matrix, check form

② $O(n)$: Size of y, check form

③ $O(n^2)$: For $\underbrace{\text{each vertex in } y}_{O(n)}$, check $\underbrace{\text{rest of list}}_{O(n)}$

④ $O(n^3)$: For $\underbrace{\text{each pair in } y}_{O(n)}$, check $\underbrace{\text{appropriate element of } x}_{O(n^2)}$

$|x| = O(n^2)$, so runtime is polynomial in $|x|$.

- If $x \in$ Yes instance, y describing path will return $M(x,y) = 1$, else if $x \in$ No, won't pass all checks $\Rightarrow M(x,y) = 0$

Knapsack: Runtime is $O(n \cdot W)$

- n = number of items
- W = size of Knapsack

What is input size?

A) $O(\log(n)\, W)$ B) $O(\log(n) + \log(W))$

C) $O(\log(n) + W)$ D) $O(n \log W)$

$X = (v_1, w_1, v_2, w_2, v_3, w_3, \ldots, v_n, w_n, W)$

need $\log_2 W$ bits

ex: $W = 2^n$. $\rightarrow |X| = O(n)$

$\rightarrow$ Runtime $= O(2^n)$

Knapsack is not known to be in P.

Yes-No (Decision) version is in NP. (Is there a way to get value v?)