Types of Problems

Easy P $O(n^c)$

(Polynomial time)

- Search
- Sort
- Matrix Mult.
- Find closest pair

1 million dollars!

$O(2^n)$ NP

Puzzles

$\boxed{\text{Sudoku}}$
Crossword

} no known polynomial alg.

Factoring large numbers

$n \stackrel{?}{=} a \times b$

Quantum

n prime?

regular

Protein Folding

Finding delivery route

Trade sequence optimization

Hard

heuristic

Chess: what is the next best move

Halting Problem

Can mathematically characterize Easy / Puzzle

$\boxed{P}$ and $\boxed{NP}$

P (Polynomial Time)

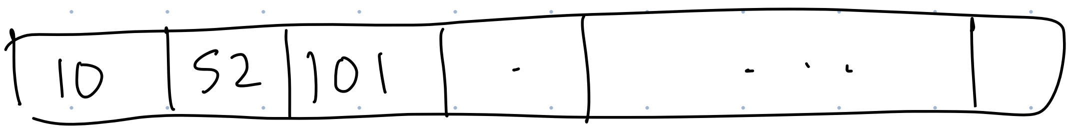
Informal: A problem is in P if it can be solved in polynomial time.

NP (Non-deterministic polynomial time)

Informal: A problem is in NP if a possible solution can be checked/verified in polynomial time.

Polynomial Time

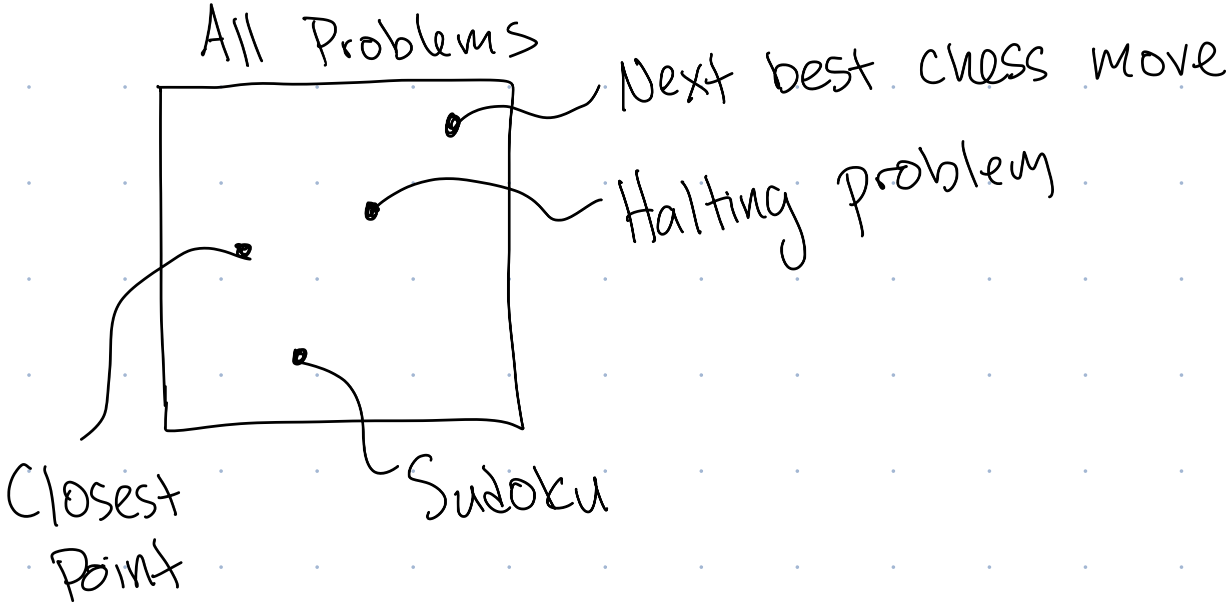
- $O(n^c)$ time for a constant c, where n = # of bits used to describe input.

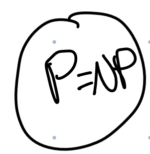


Size : m

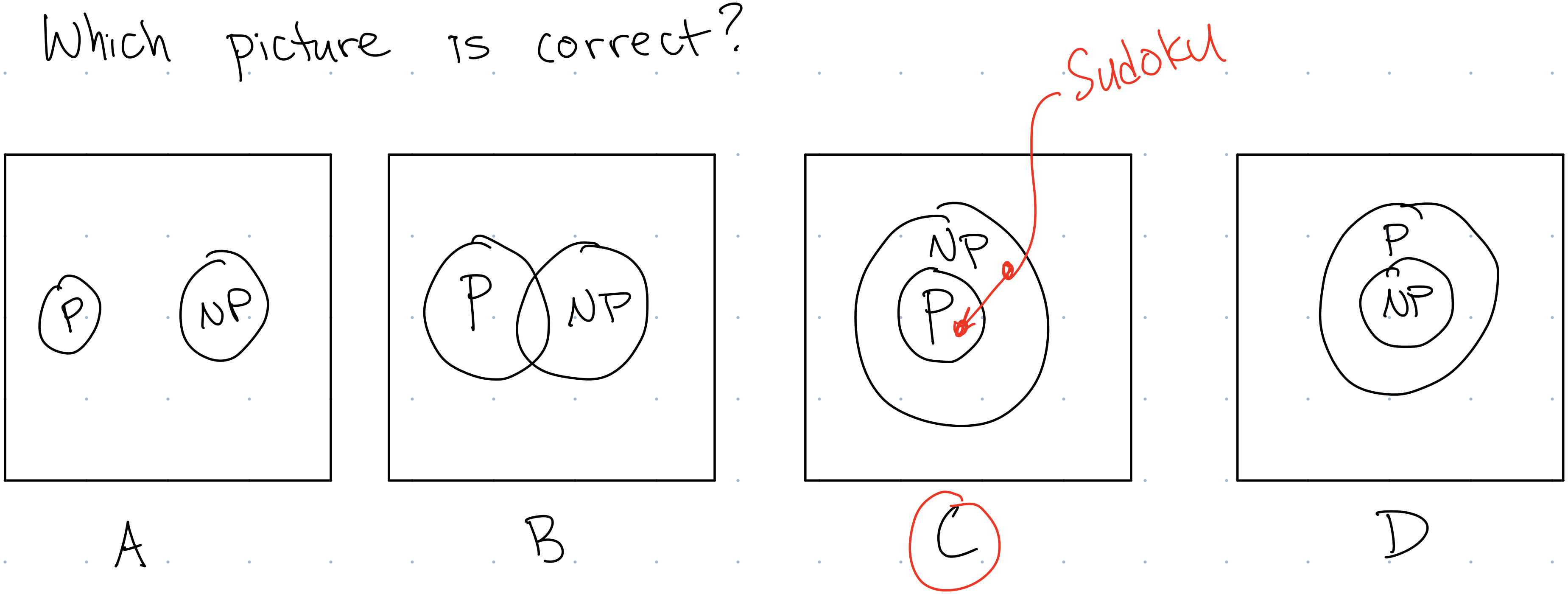
largest int : K

$n = m \log_2 K$





Which picture is correct?



name of problem

NP problems are YES-NO: $Q(x) = \begin{cases} \text{Yes} \\ \text{No} \end{cases}$

$3SAT(x) = \text{Yes}$

example of NP problem: 3SAT

3SAT : X is a Yes instance if it describes a Boolean formula that is an AND of ORs, each clause has at most 3 literals and there is a satisfying assignment.

e.g. $z_1 = T$, $z_2 = F$ ... etc. s.t. X is True

Instance: OR AND

$$X = \underbrace{(z_1 \vee z_2 \vee \neg z_3)}_{\text{Clause}} \wedge (\neg z_1 \vee \neg z_3 \vee z_4) \wedge (z_2 \vee z_4) \wedge \ldots$$

Otherwise X is a No instance

$z_1, z_2, \ldots z_n \Leftrightarrow$ variables

$z_1, \neg z_1, z_2, \neg z_2 \ldots \Leftarrow$ literals

Is $3SAT \in NP$?

Questions to Ask Yourself to Prove $Q \in NP$

① What info would convince me that x is a Yes for Q

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

② If given info from ① how could I quickly check if x is a Yes for Q

☆ You do not have to find y

☆ Only need to verify y is a solution

Proof that 3SAT $\in$ NP algorithm, instance, potential solution

- Let $M(x, y)$ be the algorithm that
  1. Check that x is an AND of OR with at most 3 literals in each clause
  2. Check y is an assignment of T, F to each variable
  3. Check that assignment in y makes x true

  And outputs 1 if all checks pass, and 0 otherwise

- $M(x, y)$ runs in $O(|x|^2)$
  1. Read through X. Size of x is $O(|x|) \rightarrow$ Time $O(|x|)$
  2. Read through y. Size of y is $|y| = O(|x|) \rightarrow$ Time $O(|x|)$
  3. For each clause, check if assignment makes it true

  $O(|x|)$ $O(|x|)$ $O(1)$ $\Big\} O(|x|^2)$

Thus 3SAT $\in$ NP

Formal(ish) Definition of NP

in $|x|$
$O(|x|^c)$

A problem is in NP iff

- YES-NO problem ("Decision 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, $\forall\, y$, $M(x,y)=0$.

witness

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

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

Show: YES instance (first matrix); NO instance (second matrix)

Prove: Hamiltonian Path $\in NP$

- Describe $M(x,y)$
- Analyze runtime of M in terms of $|x|$

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

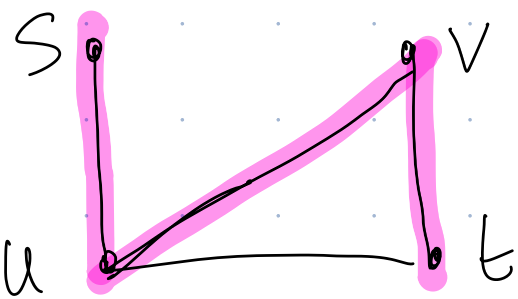
Group Work

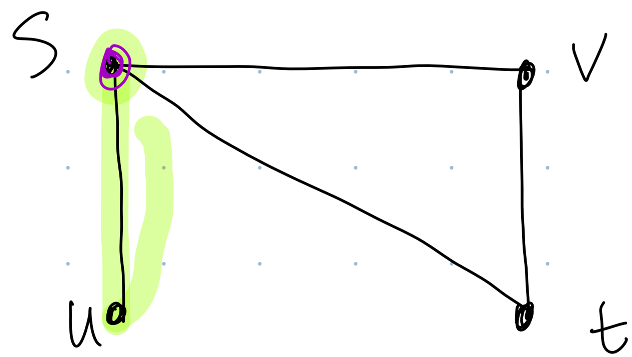
Hamiltonian Path Problem: Given an adjacency matrix for a graph $G=(V,E)$ and $s,t \in V$, is there a path from s to t that goes through each vertex once.

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

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

Show: $\Uparrow$ YES instance $\Uparrow$ No instance





Ham Path $\in$ NP (y - path from s to t)

total # of vertices

- $M(x,y)$:

① Check x is an adjacency matrix with vertices s,t. (n)

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

③ Check y has no repeated vertices

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

If passes all checks, return 1, else, 0

- Runtime: $O(n^3)$ Size of $|x|$ is n^2. $\Rightarrow |x|^{3/2} = n^3 \Rightarrow O(|x|^{3/2})$

① Check form of x $\rightarrow$ time $O(n^2)$

② Check form of y $\rightarrow$ time $O(n)$

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

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

- Thus HamPath $\in$ NP

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

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

(assume max value is \$100).

What is input size?

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

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

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

D) $\boxed{O(n \log W)}$

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

$\log_2 W$

$n \log W$ bits to write weights

$n \log V$

ex: $W = 2^n$, $n = n$ $\Big\}$ $\rightarrow |X| = O(n^2)$, $\rightarrow$ Runtime: $O(n 2^n)$

Knapsack is not known to be in P

Yes-No Knapsack $\in$ NP