9: NP

Thursday, April 15, 2021 2:33 PM

Our focus has been on P = "Polynomial Time"

P = set of problems that can be solved in polynomial time (informal definition) "easy problems"

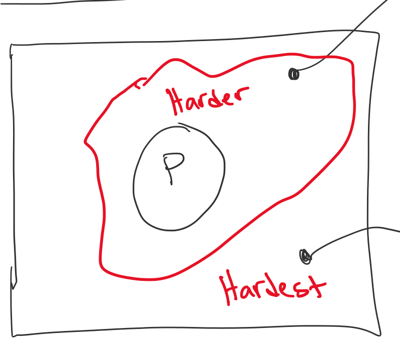
ex: Sort → $O(n \log n)$

Closest Points → $O(n \log n)$

Knapsack → $O(nC)$

... → $O(n^d)$ for d some constant

All Problems



Traveling Salesperson (Best known alg: exponential time; Unknown: if polynomial alg exists)

Halting Problem (no computer can solve)

We'd like more gradations to distinguish levels of difficulty.

- Practical: know when to stop trying to get fast alg
- Fundamental: • What is the computational power of computers?
  • Why are some problems more difficult than others?

NP = Non-deterministic Polynomial Time

A problem is in NP if

- Yes-No Problem
- ∃ M, a polynomial time algorithm s.t. (verifier)
  - If instance x has output "Yes" → $\exists y$ (witness) s.t. $|y| = O(poly(|x|))$, $M(x,y)$ outputs 1
  - If instance x has output "No" → $\forall y$ s.t. $|y| = O(poly(|x|))$, $M(x,y)$ outputs 0

Ex: 3-SAT

Instance x is a particular 3-CNF formula:

$$X = (z_1 \vee \neg z_2 \vee z_3) \wedge (z_2 \vee \neg z_4) \cdots \cdots \wedge (z_1 \vee \neg z_3 \vee z_n)$$

(clause)

X → Yes if ∃ assignment $z_i = 0$ (false) or $z_i = 1$ (true) to each literal z_i s.t. X is true

X → No if no assignment makes X true. (X can never be satisfied.)

ex: $(z_1 \vee z_2 \vee z_3) \wedge (\neg z_1 \vee \neg z_2)$ — $z_1 = 1$, $z_2 = 0$, $z_3 = 0$ } satisfies

$(z_1 \vee z_2) \wedge (z_1 \vee \neg z_2) \wedge (\neg z_1 \vee z_2) \wedge (\neg z_1 \vee \neg z_2)$ — No satisfying assignment

Q: What is |X| (the size of the instance), if it involves n literals $(z_1, \dots z_n)$.

A: $O(n)$ B: $O(n^2)$ C: $O(n^3)$ D: $O(2^n)$

8 × n choose 3 possible clauses $= O(n^3)$

(8 ↑ choice of negation)

3-SAT ∈ NP

Pf: Interpret y as an assignment

Describe verifier { Let $M(x,y) = 1$ iff:
- Each clause is true when $z_i \leftarrow y_i$.

Check verification works {
- If there is a satisfying assignment, set y to this assignment, $M(x,y) = 1$.
- If there is no satisfying assignment $M(x,y) = 0$ ∀ y.

Runtime { Runtime of M is $O(n^3)$, so polynomial in |X|.

Witness size { Size of y is $O(n)$, so polynomial in |X|.

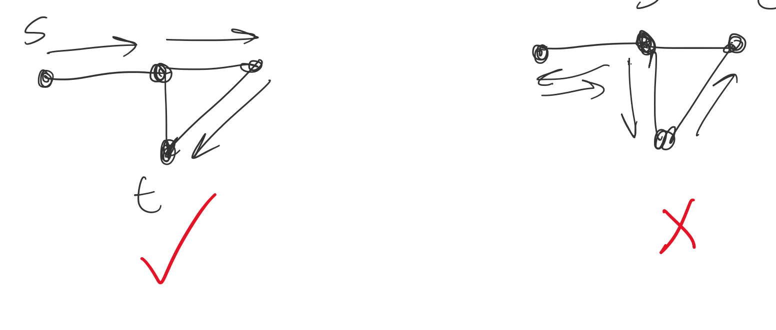
Hamiltonian Path

Instance x is description of a graph $G = (V, E)$, vertices $s, t \in V$. $|V| = n$.

X → Yes: there is a path from s to t that goes through each vertex exactly once.

X → No: No such path exists

ex:



Hamiltonian Path ∈ NP

Pf: Interpret y as sequence of edges.

Let $M(x,y) = 1$ iff: (Runtime:)
- → y is path of length n-1 (# vertices) — $O(n)$
- → y only contains edges in E — $O(n^3)$
- → y starts at s, ends at t. — $O(1)$
- → y encounters each vertex 1 time — $O(n^2)$

→ $|X| = O(n^2)$ (edges in G)

→ $|y| = O(n)$