S.KIMMEL

P, NP, and Complete Problems

Focused on problems that can be solved efficiently.

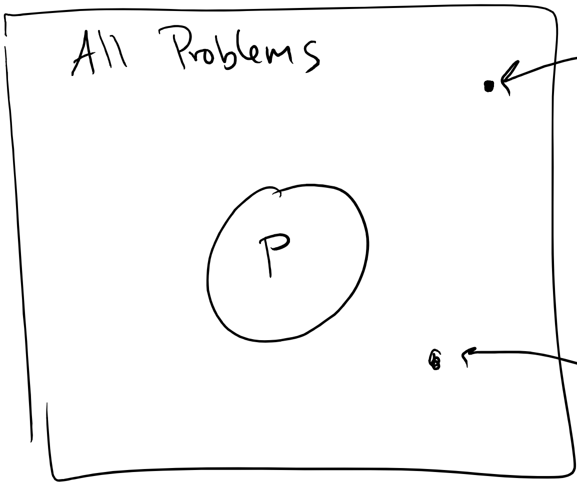
(informal) Polynomial Time

$\boxed{\begin{array}{l}P = \text{problems that can be solved in} \\ \quad O(n^k) \text{ time, } k \text{ a constant,} \\ \quad \text{if description of input is size } n \text{ (bits)}\end{array}}$

$k = 10^6$ not very efficient... but almost all problems in P actually have $k = 1, 2, 3, 4$. (like most problems in class)

e.g. Adjacency list: Size: $O((n+m)\log n)$ $\Rightarrow$ graph alg run time is $O(n+m)$, $O(nm)$, $O(n \log m)$

S.KIMMEL



Traveling Salesman (not in P, but can solve in exponential time)

Halting Problem (no computer can figure out, even with unlimited resources)

Motivates: Nondeterministic Polynomial Time

NP (informal) = set of problems where

- Solution is size $O(n^{k_1})$
- Can verify if solution is correct in $O(n^{k_2})$ time

If input is size n.

<u>ex</u>: <u>Hamiltonian Path</u>

<u>Input</u>: Adjacency List of directed, unweighted graph $G=(V,E)$; $s,t \in V$. $|V|=n$, $|E|=m$

<u>Output</u>: If it exists, a path from s to t that goes through each vertex once.

S.KIMMEL

Discuss:

- What is size of solution:

$$\underbrace{(n-1)}_{\text{\# of vertices in path}} \cdot \underbrace{\log n}_{\text{vertex nam}}$$

- What is time to verify

$O(n^2)$ ← · checking each edge is valid takes time $O(n)$
· need to do n times

$+$ $O(n)$ ← · maintain array of visited vertices to check all visited exactly once.

HAMPATH $\in$ NP