S.KIMMEL

Focus so far:

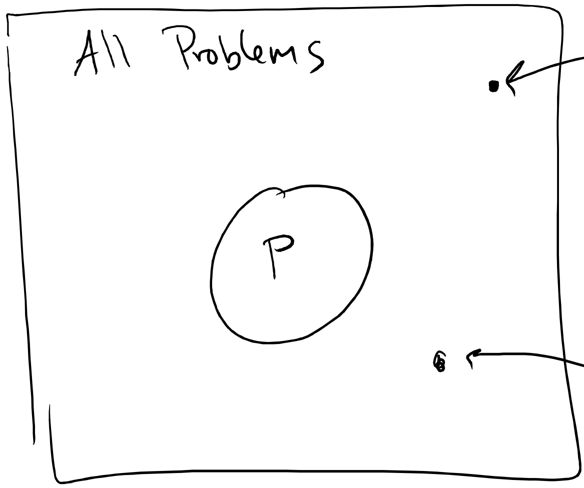
(informal) Polynomial Time

P = problems that can be solved in $O(n^k)$ time, k a constant, if description of input is size n (bits)

$k=10^6$ 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



Hamiltonian Path (probably not in P, but can solve in exponential time)

would like to distinguish between these

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

Motivates: Nondeterministic Polynomial Time

NP (informal) = set of problems where if input is size n,

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

ex: Hamiltonian Path

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

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

Another NP problem:

<u>3SAT</u>: Given CNF formula of $x_1, x_2, \dots x_n$ and negations

$\underset{\text{AND of ORs}}{\uparrow}$

where each clause has at most 3 terms, is there a satisfying assignment?

Example input:

$$\underbrace{(x_1 \vee x_3 \vee \overline{x_4})}_{C_1} \wedge \underbrace{(\overline{x_1} \vee \overline{x_2} \vee \overline{x_5})}_{C_2} \wedge \underbrace{(\overline{x_3} \vee x_5)}_{C_3}$$

Clauses: C_1 C_2 C_3

Output: $x_1 = 1$, $x_2 = 0$, ...

Most people believe NP-Hard problems take exponential time to solve

Verify HAMPATH $\in$ NP

- Size of Solution:

$$(n-1)\cdot \log n$$

↑ # of vertices in path

↖ vertex nam

- 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

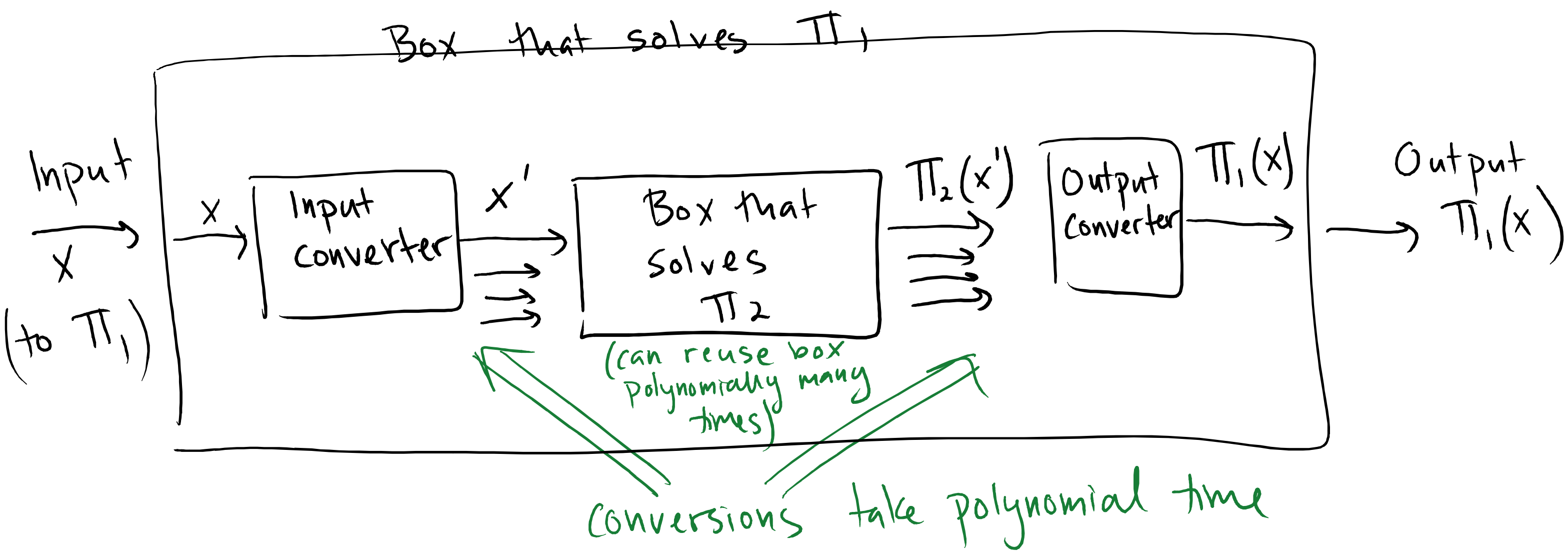
S.KIMMEL

Hamitonian cycle is hardest problem in NP

what does this mean? It can be used to solve any other problem in NP

Polynomial Reduction

Problem Π_1 reduces to problem Π_2 if



To show 3SAT reduces to HAMPATH:

→ X = CNF formula

→ Input converter uses X to create X', an Adj. List

→ Output HAMPATH(X') tell you if Ham path exists

→ Output converter uses this info to decide if CNF formula is satisfiable

S.KIMMEL

Q: If $\Pi_2 \in P$ and Π_1 reduces to Π_2 then

A) $\Pi_1 \in P$

B) $\Pi_1 \notin NP$

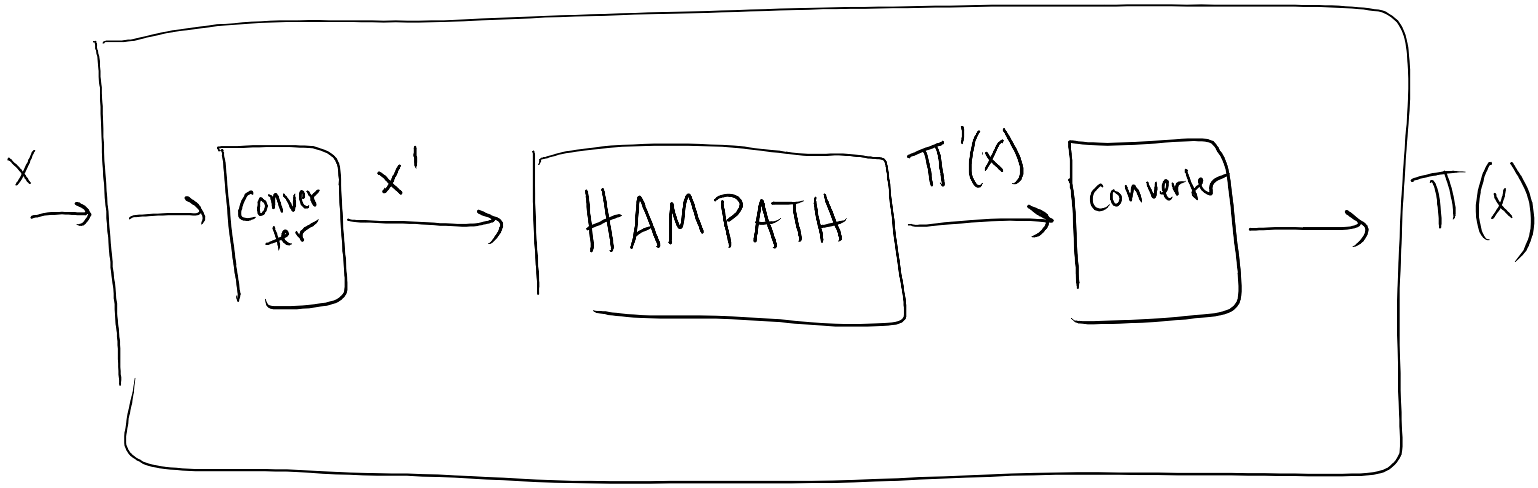
C) We don't have enough information

$\Pi_1 \in P$ because need to run Π_2 poly # of times, takes poly time, and do poly time other steps $\Rightarrow$ poly time.

Π_1 reduces to Π_2 implies Π_2 is harder than Π_1 (if can do Π_2, can do Π_1)

S.KIMMEL

Can show $\forall \Pi \in NP$, Π reduces to HAMPATH



- HAMPATH is NP-Hard (every problem in NP reduces to it)
- Also HAMPATH ∈ NP

$\Downarrow$

HAMPATH is NP-Complete

$$\boxed{\Pi \in NP \wedge \Pi \in \text{NP-Hard} = \text{NP-COMPLETE}}$$

We will take as fact that 3SAT is NP-COMPLETE

S.KIMMEL

Important Skill in algorithm design proving Π is NP-Hard

Why?

- Won't spend time trying to find efficient solution
- Look up existing alg for NP-hard problems
- Use special structure (average case might be easy)

$\boxed{\text{Strategy for proving } \Pi \text{ is NP-Hard}} \Rightarrow$ Reduce 3SAT to Π

(Π harder than 3SAT) & (3SAT harder than NP)

$\Rightarrow$ Π is harder than NP

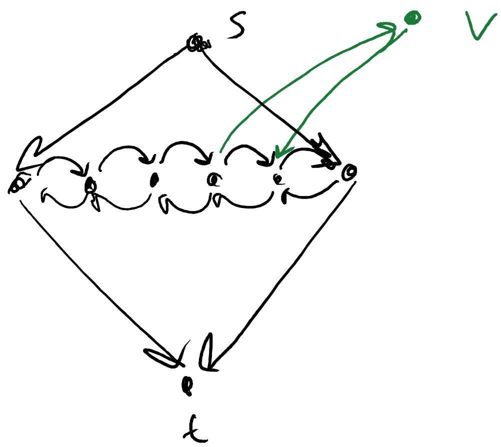
<u>Prove</u>: 3SAT reduces to HAMPATH

1) Create X' (input to HAMPATH) using X (input to 3SAT)

2) Show takes poly time to do reduction

3) Show 3SAT(X) has solution iff HAMPATH(X') has solution

S.KIMMEL



Q: How many Hamiltonian Paths are there from s to t without v / with v?

A) 1, 0

B) 1, 1

C) 1, 2

D) 2, 1 ⟸