April 16, 2024 2:21 AM

P, NP, and NP-Completeness

Which decision problems are "easy" / "hard"?

Exs

CYCLE

polytime

Input: graph G

Output: yes iff $\exists$ cycle in G

LCS:

polytime

Input: 2 sequences, number K

Output: yes iff $\exists$ common subseq of length $\geq K$

SHORTEST-CYCLE:

polytime

Input: graph G, number K

Output: yes iff $\exists$ cycle in G of length $\leq K$

LONGEST-CYCLE:

hard?

Input: graph G, number K

Output: yes iff $\exists$ (simple) cycle in G of length $\geq K$

HAMILTONIAN-CYCLE:

hard?

Input: graph G

Output: yes iff $\exists$ cycle in G that visits every vertex exactly once

VERTEX-COVER:

hard?

Input: undir graph $G = (V, E)$, number K

Output: yes iff $\exists$ subset $S \subseteq V$ of size $\leq K$ s.t. $\forall uv \in E \Rightarrow u \in S$ or $v \in S$

INDEPENDENT-SET:

hard?

Input: undir graph $G = (V, E)$, number K

Output: yes iff $\exists$ subset $S \subseteq V$ of size $\geq K$ s.t. $\forall u, v \in S \Rightarrow uv \notin E$.

CLIQUE:

hard?

Input: undir. graph $G=(V,E)$, number K

Output: yes iff $\exists$ subset $S \subseteq V$ of size $\geq K$ s.t. $\forall u,v \in S \Rightarrow uv \in E$

SET-COVER:

hard?

Input: set U of elements, collection $\mathcal{C}$ of subsets of U, number K

Output: yes iff $\exists$ subcollection $\mathcal{S} \subseteq \mathcal{C}$ s.t. union is U.

EDGE-COVER:

polytime! (CS473)

Input: undir graph $G=(V,E)$, number K

Output: yes iff $\exists$ subset of edges $S \subseteq E$ st. $\forall u \in V$, $\exists v$ with $uv \in S$.

2-COLORING:

polytime

Input: undir graph $G=(V,E)$

Output: yes iff $\exists \varphi : V \to \{red, blue\}$ s.t. $\forall uv \in E$, $\varphi(u) \neq \varphi(v)$

3-COLORING:

hard?

Input: undir graph $G=(V,E)$

Output: yes iff $\exists \varphi : V \to \{red, green, blue\}$ st. $\forall uv \in E$, $\varphi(u) \neq \varphi(v)$

3SUM:

polytime

Input: set S of n numbers, value W

Output: yes iff $\exists a,b,c \in S$ s.t. $a+b+c=W$

SUBSET-SUM:

hard?

Input: set S of n numbers, value W

Output: yes iff $\exists$ subset $T \subseteq S$ that sums to W

⋮

Def P = all decision problems that has polytime alg'ms

Def NP = all decision problems that can be expressed in the form

Input: object x
Output: yes iff $\exists$ object y s.t. some condition $R(x,y)$ is true.

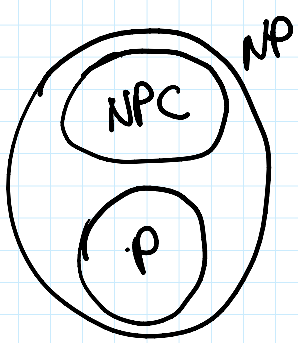
where ① object y has poly size ← certificate
② condition $R(x,y)$ can be verified in polytime.

Ex

LONGEST-CYCLE $\in$ NP:
① certificate: cycle C ← polysize
② condition to verify: $\{|C| \geq K$ & C is simple.$\}$ ← checked in polytime.

VERTEX-COV $\in$ NP:
① certificate: subset $S \subseteq V$ ← polysize
② condition to verify: $\forall uv \in E$, $u \in S$ or $v \in S$ & $|S| \geq K$. ← polytime $O(m)$

Note: NP = Nondeterministic Polytime



Fact $P \subseteq NP \subseteq$ EXPTIME ← soluable in $2^{p(n)}$ time for some polynomial for some polynomial $p(n)$.

Fact $P \subseteq NP \subseteq EXPTIME$

Pf: $(P \subseteq NP)$ ignore certificate

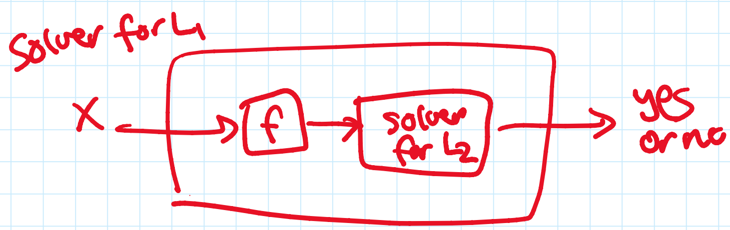
$(NP \subseteq EXPTIME)$ brute force
try all certificates $\square$

"Million-Dollar" Conjecture $\underline{P \neq NP}$.

Q: Assuming $P \neq NP$,
how to identify candidate hard problems?

idea - consider the "hardest" problem in NP

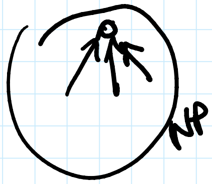
Recall: L_1 poly-time reduces to L_2
iff there is polytime computable fn f s.t.
$\forall x$, output of L_1 on x is yes
$\Longleftrightarrow$ output of L_2 on $f(x)$ is yes



iff $L_1 \leq_p L_2$

Def L is NP-hard iff $\forall L' \in NP$,
$L' \leq_p L$.

Def L is NP-complete iff ① $L \in NP$ and
② L is NP-hard.



Q: Does NPC problem exist?

Yes!

The "First" NPC Problem: Satisfiability (SAT)

Input: Boolean formula in n vars $F(x_1, \dots, x_n)$

Output: yes iff $\exists$ assignment of Boolean values to $x_1, \dots, x_n$ s.t. F evaluates to true

e.g. $F(x_1, x_2, x_3) = (x_1 \vee \overline{x_2 \vee \bar{x}_3}) \wedge \bar{x}_2$

yes $(x_1 = 1, \; x_2 = 0, \; x_3 = 0)$

($\vee$ or, $\wedge$ and)

Cook-Levin Thm (1971)

SAT is NP-complete.

Pf. Sketch: ① SAT $\in$ NP. ✓

② $\forall L \in NP$, $L \le_p$ SAT:

where L is of form:

Input: x

Output: yes iff $\exists y$ s.t. $R(x,y)$ is true

poly-time verifiable by algm $\mathcal{A}$

idea - Convert $\mathcal{A}$ into a Boolean formula

define vars z_{ij} = i^{th} bit of memory during step j.

⋮

□