NP and NP-complete

So far we have been focusing on algorithms.
What about hardness?

Example: Any comparison-based sorting algorithm must take $\Omega(n \log n)$ time.

However, there are many problems in a gray area:

- we do not know poly-time algorithms. ⟺ runtime $= O(n^c)$ for some $c = O(1)$
- we cannot prove there are no poly-time algorithms.

A set of problem that are "equivalent":
if there exists poly-time algorithm for one of them,
then there exist poly-time algorithms for all of them.

Decision vs. optimization:

Example: What is the length of the shortest path from s to t?
vs.
Is there a path from s to t with length $\le k$?
we are going to work with the decision version.

Polynomial-Time Reductions.

Reduction: a method for solving one problem using another.

Poly-time reduction: For two problems X and Y, we say $X \le_p Y$

read as "X is poly-time reducible to Y"
"X is at most as hard as Y"
(w.r.t poly-time).

iff there is a poly-time algorithm that
can transform instances of problem X (input of the reduction)
into instances of problem Y (output of the reduction) such that
the answers to both instances are always
the same.

Corollary: If X can be solved in poly-time,
then so can Y.

Example of "$\leq_p$":

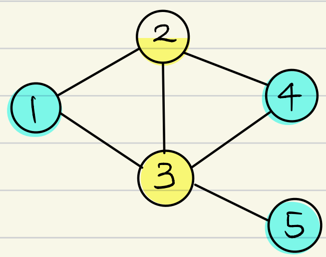
Vertex Cover: given a graph G an an integer k, decide if G has a vertex cover of size $\leq k$.

Definition (vertex cover): a set S of vertices such that every edge has at least one endpoint in S

Independent Set: given a graph H and an integer ℓ, decide if H has an independent set of size $\geq \ell$.

Definition (independent set): a set S of vertices such that there are no edges between vertices in S.

Theorem: Vertex Cover (VC) $\leq_p$ Independent Set (IS)

$G =$ 

VC(G, 2)? YES! $\{2,3\}$ is a VC.

$\Updownarrow$

IS(G, 3)? YES! $\{1,4,5\}$ is an IS

Proof: we show that S is a vertex cover if and only if $V-S$ is an independent set.

S : VC

V-S : IS

is a VC

$\Updownarrow$

every edge must have ≥ 1 end point in

$\Updownarrow$

no edge has both endpoints in

$\Updownarrow$

is an IS

$VC \leq_p IS$:

we are given a VC instance (G, k) $\longrightarrow$ $\boxed{\text{a reduction } VC \leq_p IS.\quad H \leftarrow G,\ \ell \leftarrow n-k}$ $\longrightarrow$ (H, ℓ) an IS instance that always has the same answer.

P vs. NP.

Both are complexity classes.

P: all decision problems that can be solved (deterministically). in polynomial time.

NP: all decision problems for which the "yes" instances have proofs that be verified in (deterministic) polynomial time.

Example: (the decision version of)

MST
Shortest path
Maxflow
Sorting

$\Big\} \in P$.

(the decision version of)

Vertex cover
Independent set

$\Big\} \in NP$.

Theorem: Vertex Cover $\in$ NP.

Proof: If Peggy wants to convince Victor the answer to $VC(G, k)$ is "yes"

Peggy can simply show a vertex cover of size $\leq k$.

proof : $S \subseteq V$ with $|S| \leq k$

verification : Victor can check every edge $e \in E$. to see if e has at least one endpoint in S.

Theorem: $P \subseteq NP$.

Proof:

proof $= \phi$.

verification: solve the instance and verify the answer is yes.

Open Question : $P = NP$?

NP-Complete (NPC): All decision problems X such that.

① $X \in NP$

② $\forall Y \in NP, \quad Y \le_p X$.

Intuitively. NPC contains the hardest problems in NP.

Lemma. If $X \in NPC$, $Y \in NP$, and $X \le_p Y$,
then $Y \in NPC$.

Proof. ① $Y \in NP$.

② $\forall Z \in NP, \quad Z \le_p X$ (because X is NPC).
then $Z \le_p Y$ (If $Z \le_p X$ and $X \le_p Y$, then $Z \le_p Y$)

An example of NP-Complete problems: 3-SAT.

3-SAT. Input: n boolean variables $x_1, \cdots, x_n$.
m clauses. $C_1, \cdots, C_m$.

Output: "YES" iff there is an assignment
$x_i \to \{0,1\}$
that satisfies all clauses simultaneously.

(Each clause is the OR "$\vee$"
of three literals.
Literal: x_i, $\overline{x_i}$ "not x_i".)

Example: $n=4 \quad m=3$

$C_1 = x_1 \vee \overline{x_2} \vee x_3$

$C_2 = \overline{x_1} \vee \overline{x_3} \vee x_4$

$C_3 = \overline{x_1} \vee \overline{x_3} \vee \overline{x_4}$

Answer = "YES", one solution is
$x_1 = 1$, $x_3 = 0$
x_2 and x_4 can take any value.

Problems harder than NP?

Generalized Chess $\notin$ NP.

Input: An $n \times n$ chess board position (with natural generalized chess rules)

Output: Black moves first. Output "YES" iff Black can win.