NP-Completeness:

Q: How to prove that a problem is hard to solve?
No polynomial-time algorithm.

Class P: Polynomial-Time

P = all problems that can be solved in poly-time.

eg. Shortest path, MST, LCS, interval scheduling...

independent set in any graph? Knapsack?

2-D interval scheduling?

Optimization Problem: Find a sol’n that max/min some objective

eg. Shortest path, MST, ...

up ↓ simplifies answer.

Decision Problem: Decide if there exists a sol’n with value at least / at most a given number.

eg. Jst path of length ≤ k, J MST wt weight ≥ k

J independent set of size ≥ k...
Reductions: Prob. A "reduces" to Prob. B (Decision Prob.)

Consequence: If Alg'm(B) is pol-time then Alg'm(A) is pol-time.
YES if & only if poly-time.

Why?
If \( f \) is \( O(n^c) \) then
\[
k = |f(x)| = c|x|^c.
\]
Alg'm(B) is \( O(k^d) \) for some constants \( c, d \) are constants.

\[
n = |x|\]
Total Running time of Alg'm(A) =
\[
O(n^c) + O(|f(x)|^d) = O(n^c) + O(n^{cd}) = O(n^{cd}) \text{ poly-time!}
\]

\( \neg \) (poly-time for B \( \implies \) poly-time for A).

\( \neg \) No poly-time for A \( \implies \) No poly-time for B.

Main Def: Given decision problems \( L_1, L_2 \)
a pol-time reduction from \( L_1 \) to \( L_2 \)
is a pol-time alg'm \( f \) s.t. an input \( x \) to \( L_1 \)

Output \( f(x) \) to \( L_1 \) on \( x \) is YES \( \iff \) Output \( f(x) \) to \( L_2 \) on \( f(x) \) is YES
Notation: \( L_1 \leq_p L_2 \)

**Fact 1:** If \( L_1 \leq_p L_2 \), \( L_2 \in \mathbb{E} \).

Then \( L_1 \in \mathbb{E} \implies \exists \text{ poly-time algo.} \)

**PS:** see above. (where \( A=L_1 \), \( B=L_2 \))

\( L_2 \leq_p L_3 \)

**Fact 2:** If \( L_1 \leq_p L_2 \), \( L_2 \leq_p L_3 \)

Then \( L_1 \leq_p L_3 \)

\( g \circ f \)

**PS:** By composition.

---

**Examples of Reductions:**

**Ex 1:** Vertex cover (VC): (decision ver.)

Input: Undirected graph \( G=(V,E) \), integer \( k \)

Output: YES if \( \exists \text{ vertex cover of size } \leq k. \)

\( \exists S \subseteq V \text{ s.t. } |S| \leq k \)

\( \forall u \in V \in E \implies \text{YES or NO} \)

\( D, B, F, G^3 \)
VC: \{A, C, E\}

\[ \text{Set cover (SC):} \]

Input: Set \( U, A_1, \ldots, A_n \subseteq U \), integer \( k' \)

Output: YES if \( \exists I \subseteq \{1, \ldots, n\} \) s.t. \( I \) is a set cover of size \( \leq k' \)

\[ \text{U.A}_i = U \quad \forall i \in I \]

\[ \text{eg:} U = \{1, 2, 3, 4, 5, 6\} \]

\[ \{1, 2, 3\}, \{1, 3, 5\}, \{2, 4\}, \{3, 5\} \]

\[ k' = 2 \Rightarrow \text{YES} \]

\[ k' = 1 \Rightarrow \text{NO} \]

Claim 1: VC \( \leq_p \) SC

Proof:

Give input to VC: \( G = (V, E) \), integer \( k \)

Construct input to SC: \( U, A_1, \ldots, A_n \), integer \( k' \)

where \( U = V \), \( A_i = \{ e \mid e \text{ is incident to vertex } v_i \} \)

\[ k' = k \]

Correctness:

If VC cover \( S \subseteq V \) of size \( \leq k \) in G.

\[ \Rightarrow I = \{ i \mid v_i \in S \} \] is a set cover

because

\[ U.A_i = U \{ e \mid e \text{ incident to } v_i \} \]

\[ \text{def of } A_i \Rightarrow \]

\[ \text{def of } I \Rightarrow \]

\[ \text{def of } S \Rightarrow \]

\[ \implies r = t \]
\[
\begin{align*}
&\text{x} \in E \land \forall i \in S \Rightarrow \exists e \in E \text{ incident on } v_i \land v_i \in S \\
&\text{x} \in E \land \forall i \in S \Rightarrow \exists e \in E \text{ incident on } v_i \\
&\left(\exists \text{x} \in E \land \forall i \in S \Rightarrow \exists e \in E \text{ incident on } v_i \right) \quad \left(\exists \text{x} \in E \land \forall i \in S \Rightarrow \exists e \in E \text{ incident on } v_i \right)
\end{align*}
\]

**Claim 2:** Independent set \( \leq \) V.C.

**Proof:** Given input IS: \( G = (V, E), \text{ int. } k \)
Given input to IS: $G = (V, E)$, ... 
Construct input to VC: $G'$, int. $k'$
where $G' = G$

$k' = n - k$

**Correctness pf:** I indep. set of size $\geq k'$. Say set $S$.

\[ \Rightarrow \quad \forall S \text{ is a VC } \implies \quad |S| = n - 15 \leq n - k' = k \]

**Claim 3:** $IS \leq_p VC \leq_p SC \Rightarrow IS \leq_p SC$.

How to get first hard problem?