Thursday, April 14, 2022 10:52 AM

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, ...

$\Updownarrow$ $\downarrow$ simplifies answer.

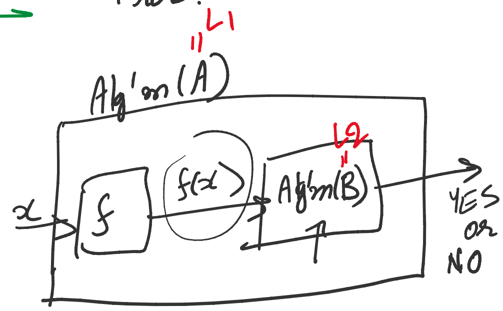
★ Decision Problem: Decide if there exists a sol'n with value at least / at most a given number.

eg. $\exists$ s-t path of length $\leq K$, $\exists$ MST of weight $\geq K'$

$\exists$ independent set of size $\geq K''$....

★ **Reductions:** Prob. A "reduces" (poly-time) to Prob. B (Decision Prob.)

Intuitions:



Consequence: If Alg'm(B) is poly-time then Alg'm(A) is poly-time?
YES iff f is poly-time.

why? If f is $O(n^c)$ time

$K = |f(x)| = c'|x|^c$.

Alg'm(B) is $O(K^d)$ time where c, d are constants.

$n = |x|$

Total Running time of Alg'm (A) =

$$O(|x|^c) + O(|f(x)|^d)$$

$$= O(n^c) + O((c'n^c)^d) = O(n^c) + O(n^{c \cdot d})$$

$$= O(n^{c \cdot d}) \quad \text{poly-time!}$$

$\neg$ (Poly-time for B $\Rightarrow$ poly-time for A.)

$\equiv$ No poly-time for A $\Rightarrow$ No poly-time for B.

Main Def: Given decision problems L_1, L_2 a poly-time reduction from L_1 to L_2 is a poly-time alg'm $\underline{f}$ s.t. $\forall$ input $\underline{x}$ to L_1

$$\boxed{\text{Output of } L_1 \text{ on } x \text{ is YES} \iff \text{Output of } L_2 \text{ on } f(x) \text{ is YES}}$$

Notation: $L_1 \le_p L_2$

Fact 1: If $L_1 \le_p L_2$, & $L_2 \in P$.

then $L_1 \in P \longrightarrow \exists$ poly-time algo.

Pf: see above. (where $A = L_1$, $B = L_2$)

Fact 2: If $L_1 \overset{f}{\le_p} L_2$, $L_2 \overset{g}{\le_p} L_3$

then $L_1 \underset{g \circ f}{\le_p} L_3$

Pf: By composition.

Examples of Reductions:

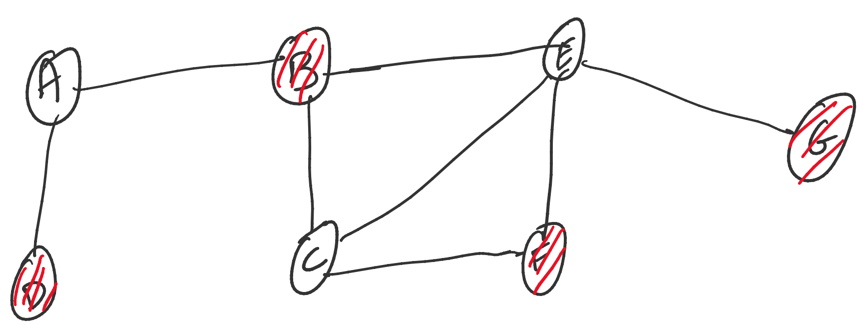
Ex1: Vertex cover (VC): (decision ver)

Input: Undirected graph $G = (V, E)$, integer K

Output: YES iff $\exists$ vertex cover of size $\le K$.

iff $\exists S \subseteq V$ s.t. $|S| \le K$

$\forall uv \in E \Rightarrow u \in S$ or $v \in S$



IS: $\{D, B, F, G\}$

IS: $\{D, B, F, G\}$

VC: $\{A, C, E\}$

$\rightarrow$ Set Cover (SC):

Input: Set U, $A_1, \dots A_n \subseteq U$, integer k'

Ouput: YES iff $\exists$ set cover of size $\leq k'$

iff $I \subseteq \{1, \dots n\}$ s.t. $|I| \leq k'$

$$\bigcup_{i \in I} A_i = U$$

eg $U = \{1, 2, 3, 4, 5\}$

$\{1,2,3\}$, $\{1,3,5\}$, $\{2,4\}$, $\{3,5\}$

$k' = 2 \rightarrow$ YES

$k' = 1 \rightarrow$ NO.

Claim I: Vertex cover $\leq_p$ Set cover.

pf: Given input to VC: $G = (V, E)$, $V = \{v_1, \dots, v_n\}$, integer k.

f: Construct input to SC: U, $A_1, \dots A_n$, integer k'

where $U = E$, $A_i = \{e \mid e \text{ is incident to vertex } v_i\}$

$k' = k$

Correctness: $\exists$ Vertex cover $S \subseteq V$ of size $\leq k$ in G.

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

because

$$\bigcup_{i \in I} A_i \overset{(\because \text{def } A_i\text{'s})}{=} \bigcup_{i \in I} \{e \mid e \text{ incident on } v_i\}$$

$$(\because \text{def. of } I) = \bigcup_{v_i \in S} \{e \mid e \text{ incident on } v_i\}$$

$v_i \in S$

$$(\because \text{def. of VC}) = E = U$$

$$(\because \text{def of } U)$$

$$\Leftarrow \quad \bigcup_{v_i \in S'} \{e \mid e \text{ incident on } v_i\}$$

$$(\because \text{def of } I) = \bigcup_{i \in I} \{e \mid e \text{ incident on } v_i\}$$

$$(\because \text{def of } A_i\text{'s}) = \bigcup_{i \in I} A_i$$

$$(\because I \text{ is SC}) = U$$

$$(\because \text{def of } U) = E$$

$\square$

Ex2: Independent set:

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

Output: YES iff $\exists$ independent set of size $\geq k$

iff $\exists S \subseteq V$ s.t. $|S| \geq k$

$$\forall u, v \in S \Rightarrow uv \notin E$$

contrapositive: $\forall uv \in E \Rightarrow u \notin S$ or $v \notin S$

$$\Rightarrow u \in V \setminus S \text{ or } v \in V \setminus S$$

$$\equiv$$

$V \setminus S$ is a V.C.

Claim 2: Independent set $\leq_p$ V.C.

Pf: Given input to IS: $G = (V, E)$, int. $\underline{k}$.

VC: G', int k'

Pf: Given input to IS: $G=(V,E)$, ...

Construct input to VC: G', int. k'

where $G' = G$

$k' = n - k$

Correctness pf: $\exists$ indep. set of size $\geq k'$

say set S

$\Longleftrightarrow$ $V \setminus S$ is a VC &

$|V \setminus S| = n - |S| \leq n - k' = k$

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

How to get first hard problem?