Lecture-22 (7/11/24)

Among all the problems in NP, what are the 'hardest' problems?

NP-Complete problems:

A problem X is NP-Complete if and only if
1) $X \in NP$
2) for all $Y \in NP$, $Y \leq_p X$

Theorem: If X is NP-Complete, then X is in P if and only if P = NP

3-SAT is NP-Complete $\Rightarrow \forall Y, Y \in NP$
$Y \leq_p$ 3-SAT — (1)

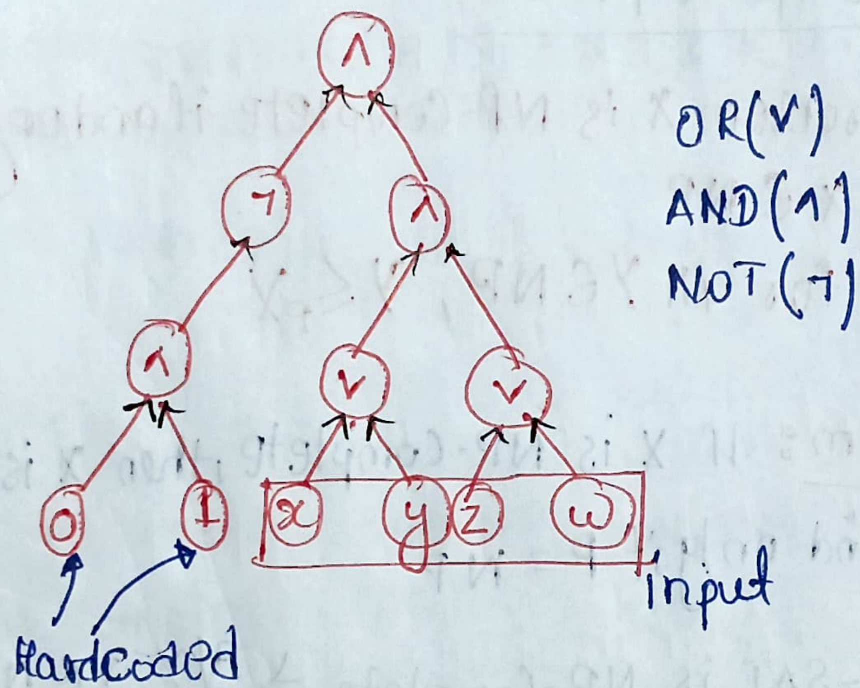
We want to prove X is NP-Complete

(1) Prove $X \in NP$
(2) $3SAT \leq_p X$

What is the first NP-Complete problem?

Circuit Satisfiability:

What is a circuit?



- A circuit K is a labeled directed acyclic graph.
- Circuit K satisfies the following properties:

Property 1: source: Constants (0/1) or input labeled as distinct variables

Property 2: Every other node is labelled as ∨ ∧ ¬

Property 3: There is a single node with no outgoing edge representing the output node

Circuit Satisfiability problem:
We are given a circuit as input, we need to decide whether there is an assignment of values to the circuit input that causes the circuit output to take value 1.

COOK-LEVIN Theorem: Circuit Satisfiability problem is NP-Complete

1. Circuit Satisfiability $\in$ NP
2. Consider any $X \in NP$, then $X \leq_p$ Circuit Satisfiability

→ The transformation from an algorithm to a circuit is the proof of COOK-LEVIN theorem

→ $X \in NP \rightarrow X$ has an efficient certifier algorith $B(s, t)$

Input to 'x'

t is the certificate

→ $s \in X$ for a fix length $|s| = n$ iff $\exists t$, $t = P(|s|)$ and $B(s,t) \doteq$ YES

COOK-LEVIN Reduction:

$s \in X$ if and only if there is a way to set the input bits to circuit K (where K is the circuit corresponding to algorithm B for fixed input length) so that circuit

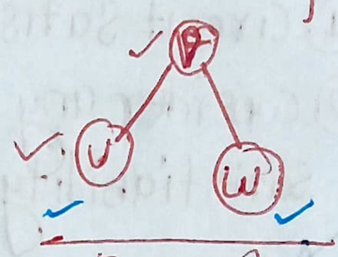
Produces output 1 (x is satisfiable)

<u>Problem:</u> Given a graph G, does it contain an independent set of size atleast 2?

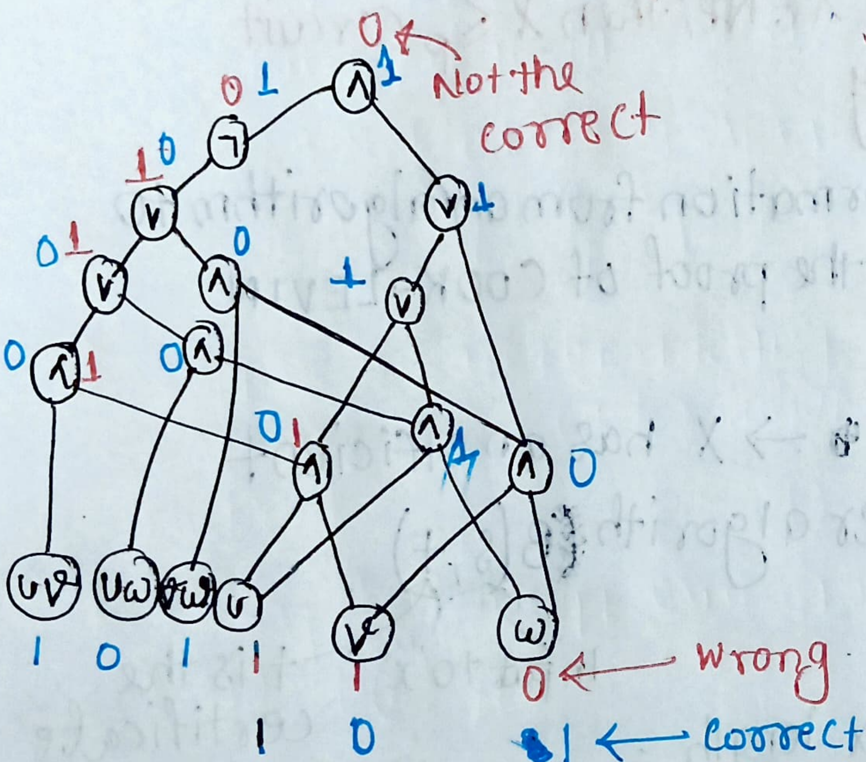
Fix the input size, $n = 3$ $V = \{u, v, w\}$

$E = \{uv, vw, uw\}$

Possible



Example



Circuit Satisfiability $\leq_p$ SAT $\leq_p$ 3-SAT

$\leq_p$ Independent set

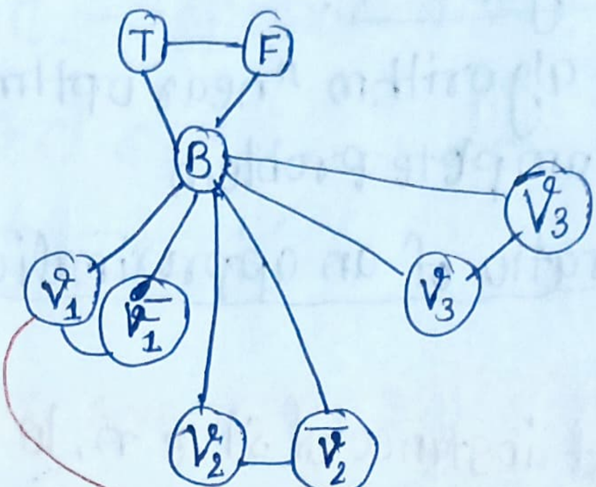
$\leq_p$ Vertex cover

$\leq_p$ Set Cover

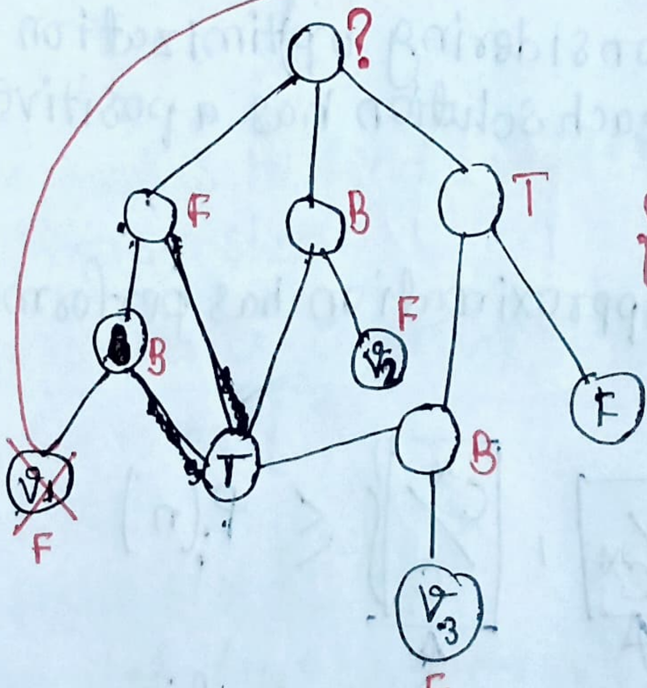
(Vertex coloring)

Theorem: $3\text{-SAT} \leq_p 3\text{-Coloring}$

Gadget 1:



$C_1 = (x_1 \vee \overline{x}_2 \vee x_3)$



$\{ F, T, B$

How to approach NP complete problems as an algorithm designer?

Approximation Algorithm:

Approximation algorithm "near optimal" solution for an NP-Complete problem

- performance ratio of an approximation algorithm:

For input instance of size n, let

C: cost of the approximation algorithm

C^*: $\boxed{\text{cost}}$ of the optimal solution

- Suppose we are considering optimization problems whose each solution has a positive cost

- We say that an approximation has performance ratio $P(n)$ if

$$\max\left\{\boxed{\frac{C}{C^*}}, \boxed{\frac{C^*}{C}}\right\} \leq P(n)$$

Minimization problem　　Maximation problem

Maximization problem

C^* — optimal
C — Approximation

$C \leq C^*$ $\qquad$ $C^* \leq C$

Minimization problem

Vertex cover: A vertex cover C of an undirected graph $G(V, E)$ is a subset $C \subseteq V$ such that if $(u, v) \in E$ the either $v \in C$ or $u \in C$ or both. The size of the vertex cover $|C|$ is the number of vertices in it.

Our goal is to find a vertex cover C with minimum size.

Approx-Vertex-Cover$(G(V, E))$

1. $C = \phi$ or $\{ \}$
2. $E' = E$
3. while $E' \neq \{ \}$ or ϕ
4. Let (u, v) be any arbitrary edge of E'
5. $C = C \cup \{u, v\}$
6. Remove from E' every edge incident either on v or u
7. Return C

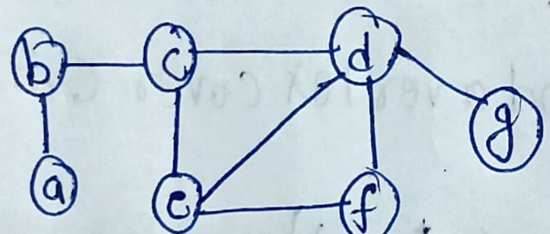
C^* be optimal vertex cover

- (u, v) the first edge picked by Approx-Vertex-Cover.
  - C* must contain atleast one of u or v

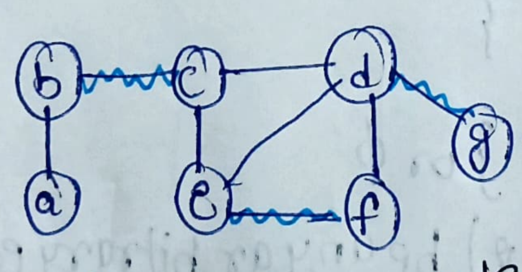
(u', v') the second edge picked by Approx-vertex-cover

{(u', v') are vertex disjoint from (u, v)}

- Any solution must contain one of the ~~edges~~ endpoints of edges picked by Approx-Vertex-cover

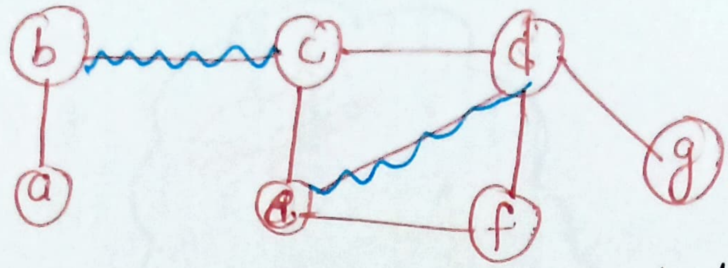


Solution 1:



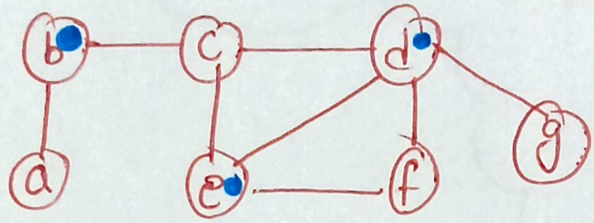
$C = \{b, c, e, f, d, g\}$ $\frac{|C|}{|C^*|} = \frac{6}{3} = 2$

Solution 2:



$C = \{b, c, e, d\}$ $\frac{|C|}{|C^*|} = \frac{4}{3} = 1.\overline{3}$

Optimal:



$C^* = \{b, e, d\}$

Theorem:

Approx-Vertex-Cover$(G(V,E))$ is a polynomial time 2-approximation algorithm:

Proof: Let A denote the set of edges seelected by the algorithm. All these edges are vertex disjoint therefore any optimal solution to vertex cover problem must contain atleast one endpoint of every edge in A.

Let C^* be an optimal solution

$|C^*| \geq |A|$ —— ①

Let C be the solution produced by the algorithm $|C| = 2|A|$ —— ②

$\therefore |C| = 2|A| \leq 2|C^*|$

① and ② $\Rightarrow |C^*| \geq \frac{|C|}{2}$ or $\frac{|C|}{|C^*|} \leq 2$ proved.

⚠

Page1-5 (L22)
Not important from Exam POV