P Versus NP Complexity Theory

- Anuja Nagare

Framework for research on Exp time problems

- NP-Hard and NP-Complete

- Write polynomial time non-deterministic algorithm

- Solve / Try to relate problems :
  - Show similarity/relation/association between problems

Reference link: https://www.youtube.com/watch?v=e2cF8a5aAhE

Nondeterministic

```
Algorithm NSearch(A,n,key)
{
        j=choice();
        if(key=A[j])
        {
            write(j);
            Success();
        }
        write(0);
        Failure();
}
```

Solve / Try to relate problems

- Step 1 : find a Base NP hard problem

- Step 2:
  - Try to relate your NP hard problem with base NP hard problem using reduction

Satisfiability $\leq_p$ 0/1 Knapsack problem

- Show that : Satisfiability reduces to 0/1 Knapsack problem
  - Satisfiability $\leq_p$ 0/1 Knapsack problem
  - Steps :
    - Take 1 example CNF Satisfiability formula
    - Take 1 example of 0/1 Knapsack problem
    - Show how similar technique can be used to solve CNF Satisfiability and 0/1 Knapsack problem
    - Then state : Both problems can be solved in similar way so both problems are NP Hard
    - As satisfiability problem has a non-deterministic algorithm thus, it belongs to NP-complete class
    - So, we can write a non-deterministic algorithm for 0/1 Knapsack problem, thus it belongs to NP complete class

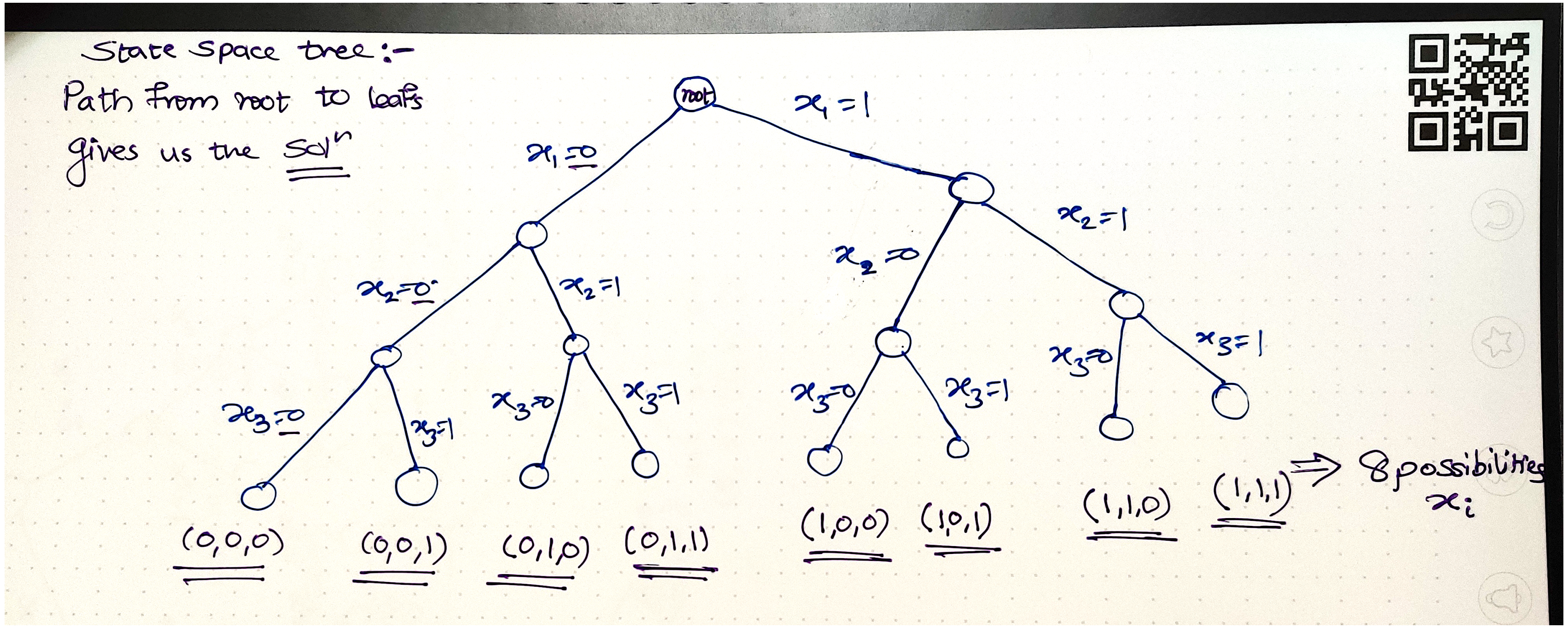
- **Base Problem : CNF Satisfiability**
- x_i = $\{x_1, x_2, x_3\}$ → Boolean variables

- Eg. CNF formula => $\underbrace{(x_1 \vee \overline{x_2} \vee x_3)}_{Clause_1} \wedge \underbrace{(\overline{x_1} \vee x_2 \vee \overline{x_3})}_{Clause_2}$

- Satisfiability problem is to find out for what values of x_i given formula is true

Possible values of x_i $\rightarrow$ 8 values $\rightarrow$ 2^3

x_1	x_2	x_3
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



0/1 Knapsack Problem

Given weights and profits of n items, Put these items in a knapsack of capacity W to get the maximum total value in the knapsack.

- Given
  - Two integer arrays : Profit [0..n-1] and weight[0..n-1] which represent values and weights associated with n items respectively.
  - Weight which represents knapsack capacity,
- Find out :
  - Maximum profit subset of profit[] such that sum of the weights of this subset is smaller than or equal to W.

You can’t break an item, either pick complete item or don’t pick it (that’s 0/1 property).

Knapsack 0-1 Problem

- The goal is to **maximize the value of a knapsack** that can hold at most W units (i.e. lbs or kg) worth of goods from a list of items I_0, I_1, … I_{n-1}.
  - Each item has 2 attributes:
    1) Value – let this be v_i for item I_i
    2) Weight – let this be w_i for item I_i

$$P = \{ \overset{p_1}{10}, \overset{p_2}{8}, \overset{p_3}{12} \} \qquad n = 3$$

$$w = \{ \overset{w_1}{5}, \overset{w_2}{4}, \overset{w_3}{3} \} \qquad \boxed{m = 8}$$

$$\quad x_1 \quad x_2 \quad x_3$$

$(0/1; 0/1; 0/1)$

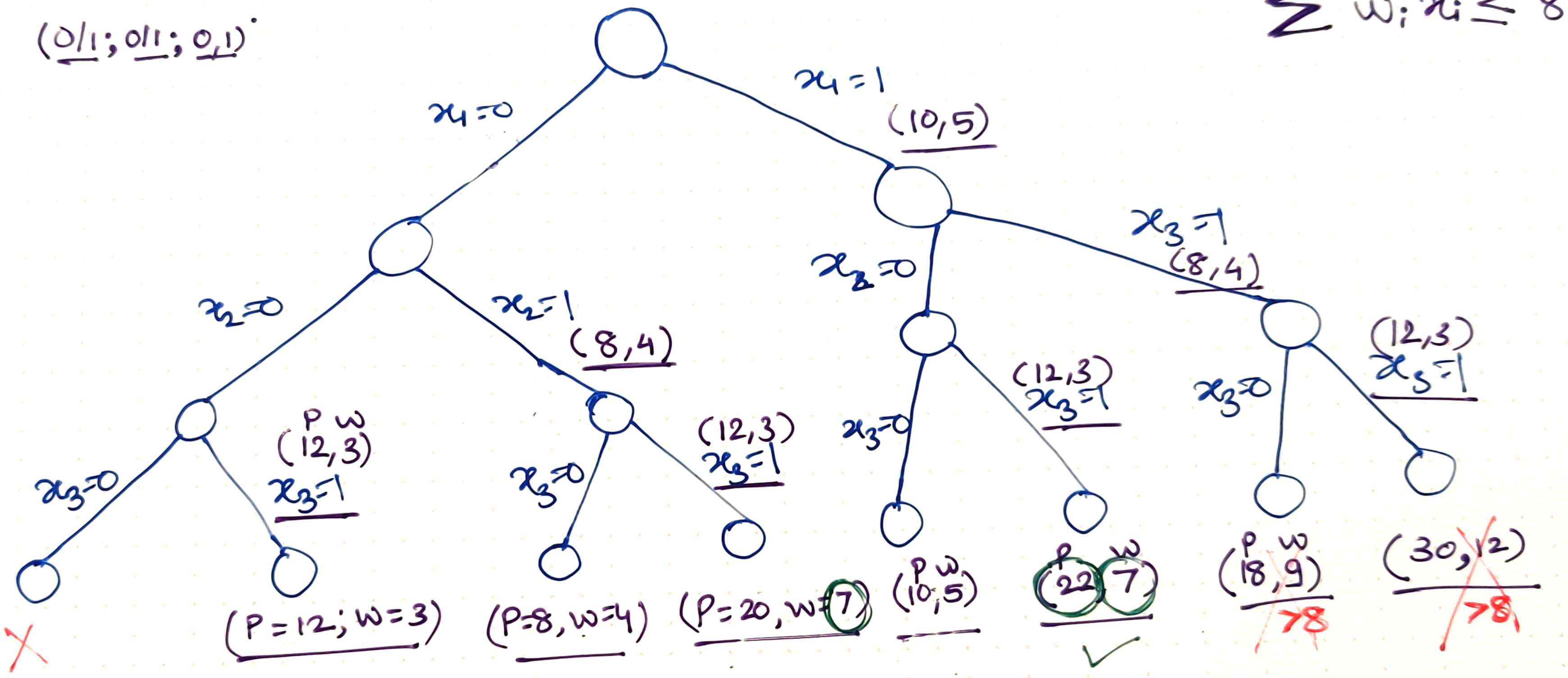
0 → dropped the item x_i

1 → Picked the item x_i

Goal

$$\max \sum p_i x_i$$

$$\sum w_i x_i \le 8$$



Solution

max profit = 22

max possible wt = 7

$$\begin{pmatrix} x_1 & x_2 & x_3 \\ 1 & 0 & 1 \end{pmatrix}$$

Mapping 0/1 Knapsack problem like Satisfiability problem

Homework :
(try to solve in similar way by mapping 0/1 Knapsack problem like Satisfiability problem)

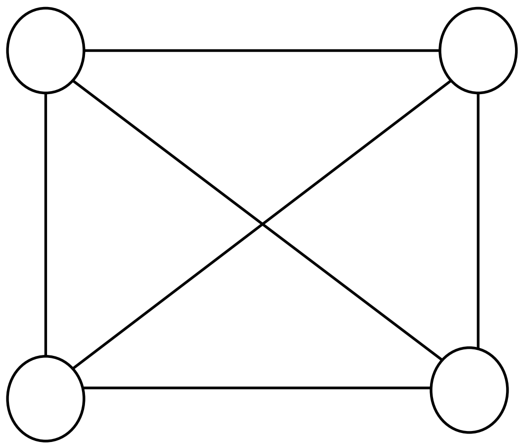
0/1 Knapsack Problem:

Weight= {2,3,4}
Profit = {1,2,5}
Knapsack max capacity weight **M=6**

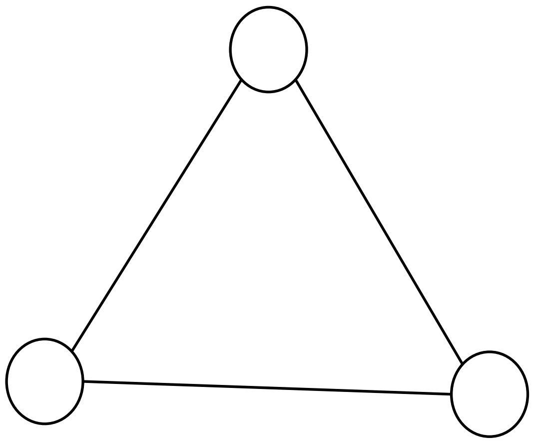
Goal:
max $\sum p_i x_i$; $\sum w_i x_i \leq M$

Complete Graph

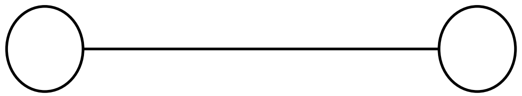
- From every vertex there is an edge connecting to every other vertex.
- Property :
  - | V| = n
  - | E | = n (n-1)/ 2



V=4 ; E = 4*(3) /2 = 12/2 = 6



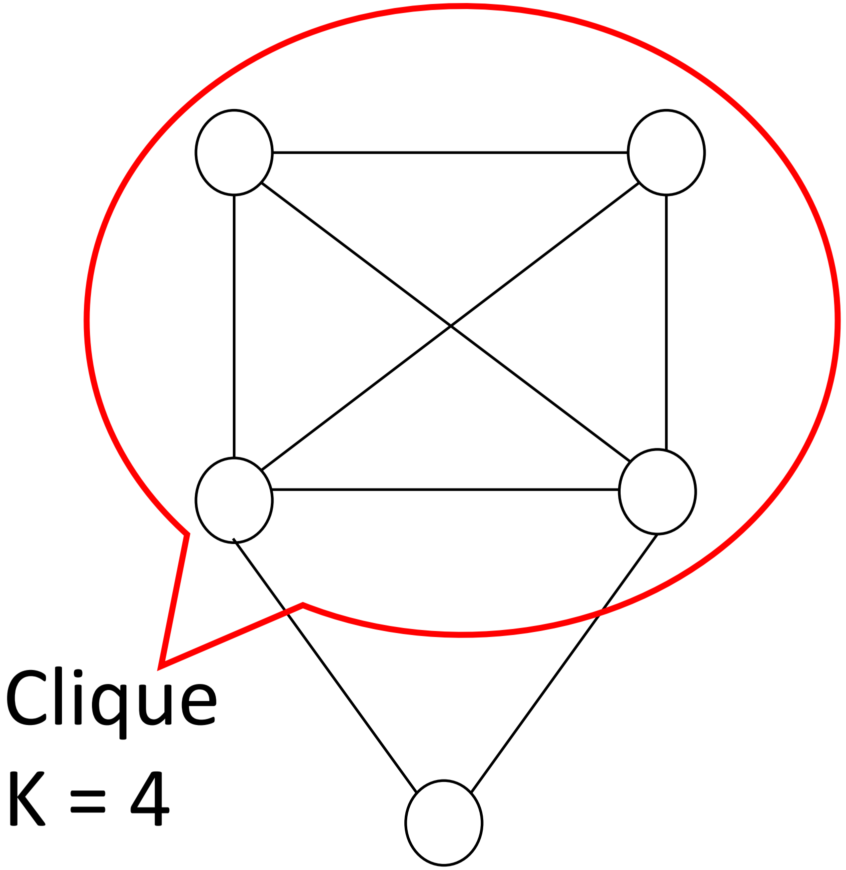
V=3 ; E = 3 (2)/2 = 6/2 = 3

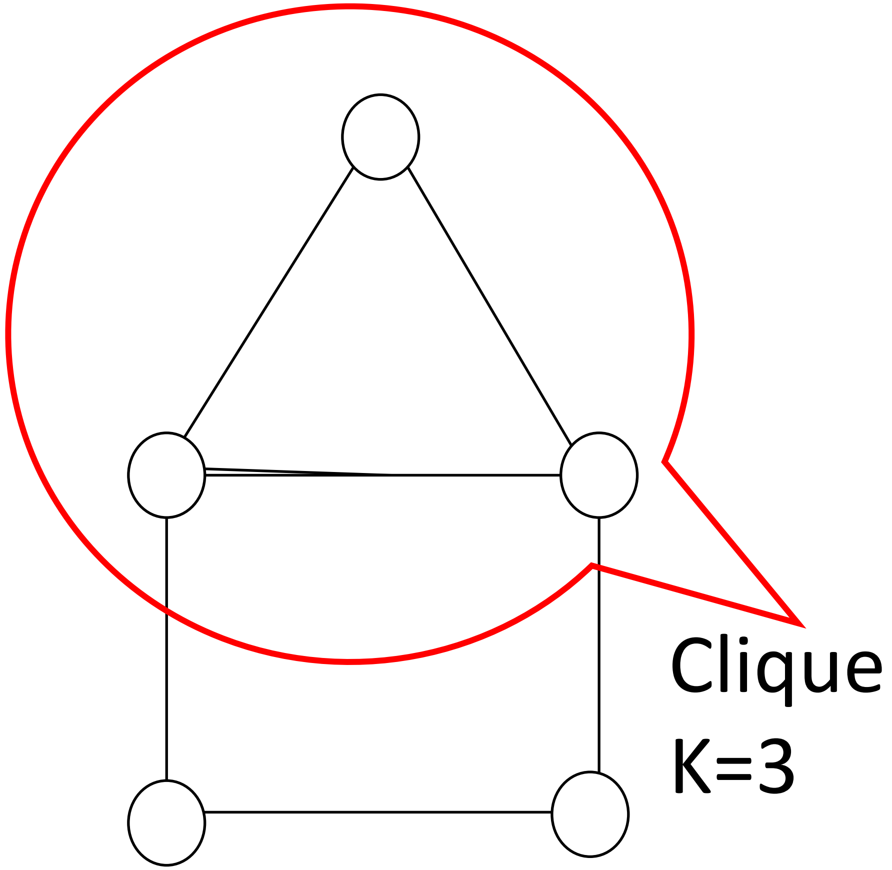


V=2 ; E = 2 (1)/2 = 1

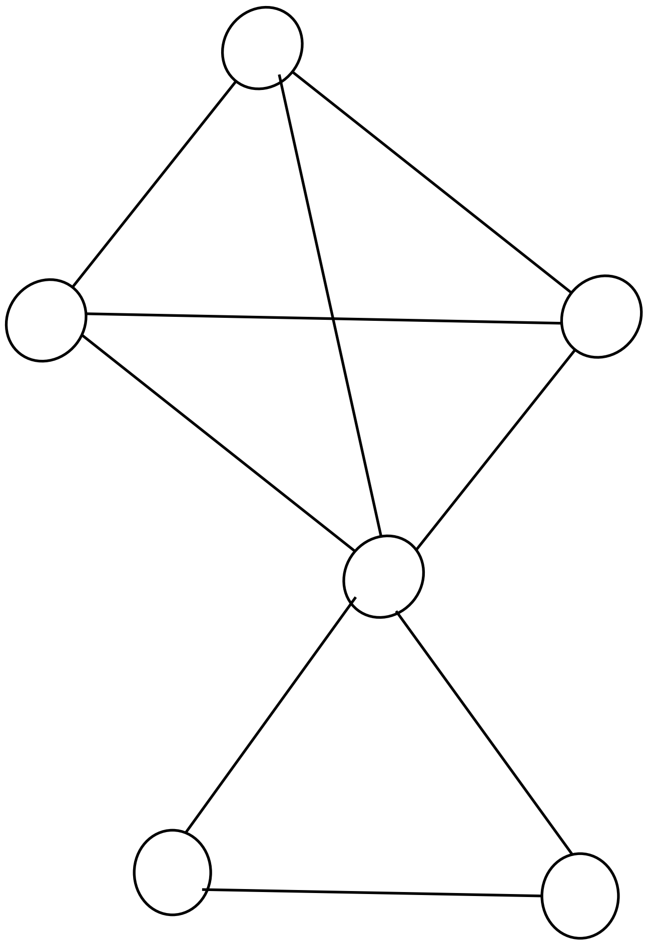
Clique Decision Problem

- Subgraph of a complete graph





Clique Decision Problem



- Does it have a clique of size 4 ?
- Does it have a clique of size 3 ?
- Does it have a clique of size 2 ?

- When you can have yes/ no type answer for a Problem it's called Decision Problem

- Optimization problem :
  - What is the maximum clique size , answer => k =4,

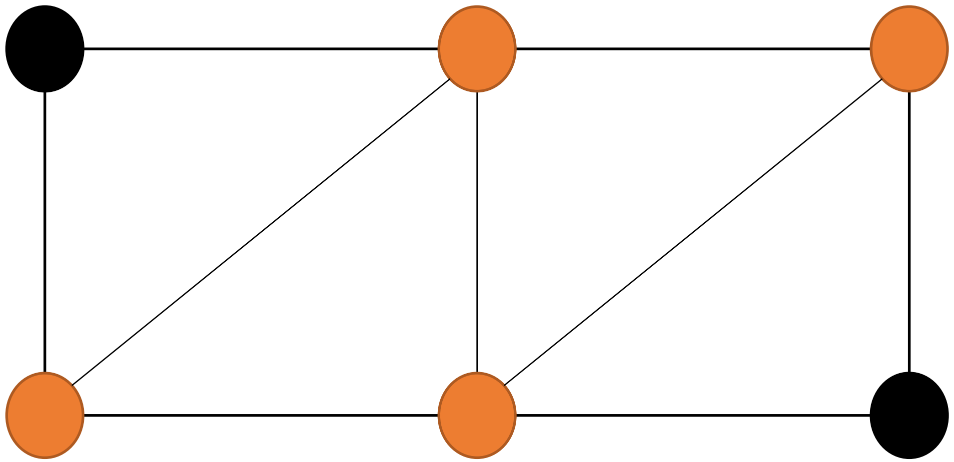
Satisfiability $\leq_p$ Clique Decision Problem

- Show that : Satisfiability reduces to Clique Decision Problem
  - Satisfiability $\leq_p$ Clique Decision Problem
  - Steps :
    - Take 1 example CNF Satisfiability formula
    - Take 1 example of 0/1 Clique Decision Problem
    - Show how similar technique can be used to solve CNF Satisfiability and Clique Decision Problem
    - Then state : Both problems can be solved in similar way so both problems are NP Hard
    - As satisfiability problem has a non-deterministic algorithm thus, it belongs to NP-complete class
    - So, we can write a non-deterministic algorithm for Clique Decision Problem, thus it belongs to NP complete class

Solve problem : Whiteboard

Vertex cover

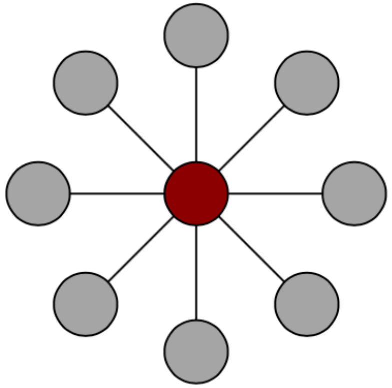
- Def:
  - Undirected graph G = (V,E)
  - S ⊆ V is a vertex cover
    - If every edge is incident on a vertex in S



- Proposition:
  - Given a graph G, the size of a matching in G Is at most the size of a vertex cover

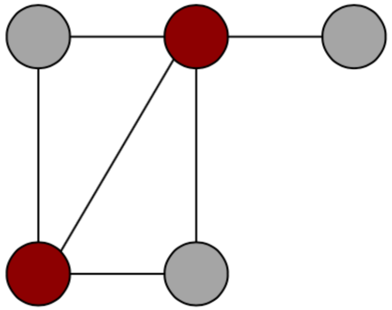
Vertex Cover

Given $G=(V,E)$, find the smallest $S \subseteq V$ s.t. every edge is incident to vertices in S.



Decision Version:

Given a graph and a number k, does the graph contains a vertex cover of size at most k?



Satisfiability $\leq_p$ Vertex cover

- Show that : Satisfiability reduces to Vertex cover Problem
  - Satisfiability $\leq_p$ Vertex cover
  - Steps :
    - Take 1 example CNF Satisfiability formula
    - Take 1 example of 0/1 Vertex cover Problem
    - Show how similar technique can be used to solve CNF Satisfiability and Vertex cover Problem
    - Then state : Both problems can be solved in similar way so both problems are NP Hard
    - As satisfiability problem has a non-deterministic algorithm thus, it belongs to NP-complete class
    - So, we can write a non-deterministic algorithm for Vertex cover Problem, thus it belongs to NP complete class

Solve problem : Whiteboard