omsairam omsrivenkatesayanamaha ommahaganadhipatayanamaha omnamahashivayaha

Clique Decision Problem (CDP):

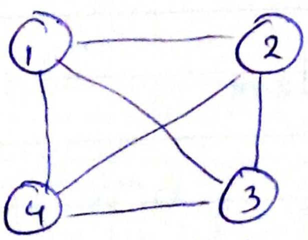
1. what is a clique?
2. what are Decision and Optimization problems?
3. what are NP-Hard Graph problems?
4. Procedure to prove NP-Hard
5. prove CDP is NP-Hard.

Complete Graph:-

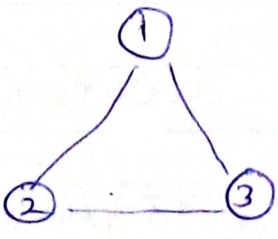
- A graph in which every node has an edge to every other node in the graph is called "Complete Graph"
- If you consider, # of vertices $= n$, then

# of edges $= \frac{n(n-1)}{2}$

Examples:-



- complete graph of '4' vertices



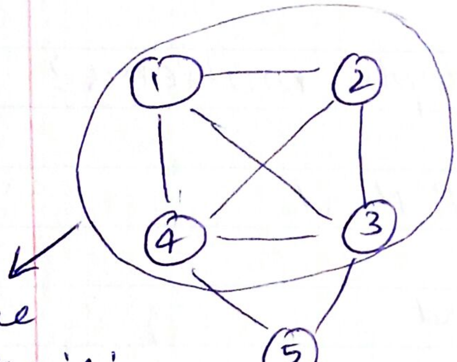
- C.G. of '3' vertices

- C.G. of '2' vertices

Clique:-

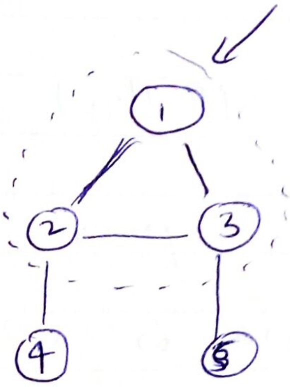
A subgraph of a graph, where the sub-graph is a complete graph is called clique.

Ex:-

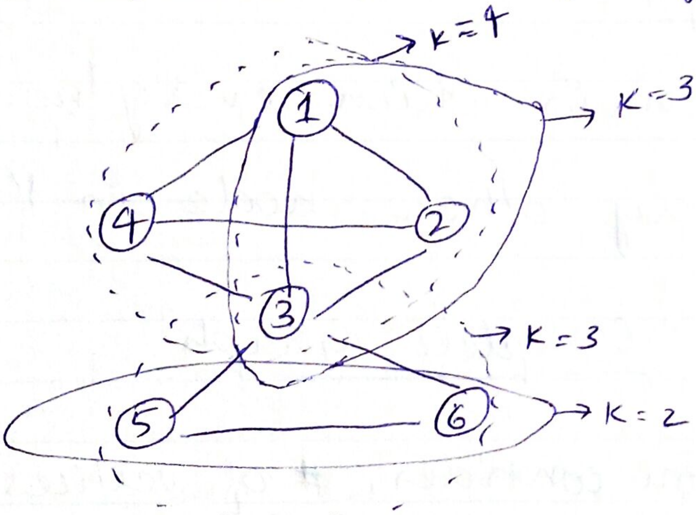


clique of size '4'.

clique of size '3'.



Identify cliques in following graph:



Decision Problem:

Problem which requires answer as "Yes" or "No" is a decision Problem.

Ex: Is there a 'clique' of size = 2 in the above problem?

Optimisation Problem:

What is the maximum size of the clique in the graph?

Reducing 3-SAT to Clique: Example:

literals = x_1, x_2, x_3

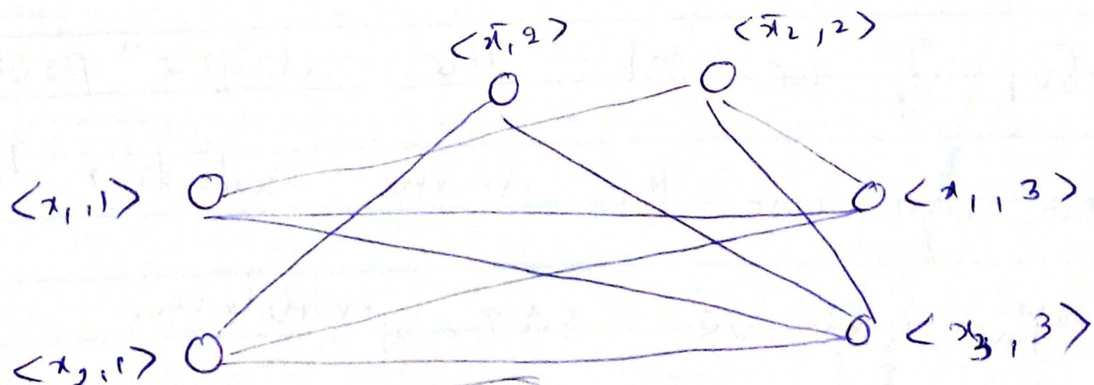
3-CNF SAT formula, $F = \underbrace{(x_1 \vee x_2)}_{c_1} \wedge \underbrace{(\bar{x}_1 \vee \bar{x}_2)}_{c_2} \wedge \underbrace{(x_1 \vee x_3)}_{c_3}$

- Draw a graph using below rules

$V = \{ \langle a, i \rangle \mid a \in C_i \}$

$E = \{ \langle a, i \rangle, \langle b, j \rangle \mid i \neq j \text{ and } b \neq \bar{a} \}$

- No connections in the same clique
- No connection b/w x_1 & $\bar{x}_1$

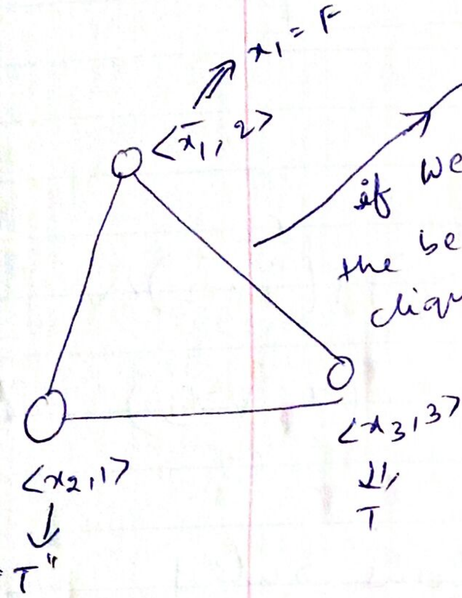


$\boxed{K = 3}$

To find the ~~clique size~~ variable assignment values, select a clique of maximum size, and mark the corresponding ~~literal~~ variable values ~~as True, if it is present in the clique~~ accordingly.

x_1	x_2	x_3
F	T	T

These two nodes are



if we select the below clique. The above one is the table

→ If we fill the above table values in the given formula,

$$F = \underset{F \quad T}{(x_1 \vee x_2)} \wedge \underset{T \quad F}{(\bar{x}_1 \vee \bar{x}_2)} \wedge \underset{F \quad T}{(x_1 \vee x_3)}$$

$$C_1 = T \qquad C_2 = T \qquad C_3 = T$$

$$= T$$

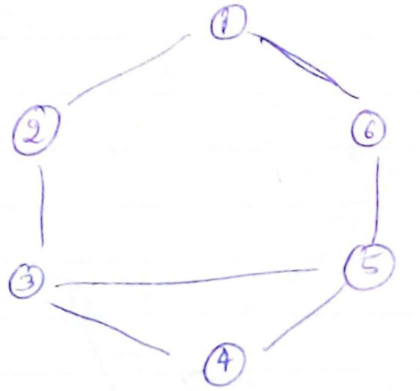
⊙ So, if we solve the "clique" problem, we can use the same solution to solve the above SAT problem.

Minimum Spanning Tree :-

- No. of spanning trees for a graph

$$= \left({}^{|E|}C_{|V|-1} - \text{no. of cycles in graph} \right)$$

Ex :-



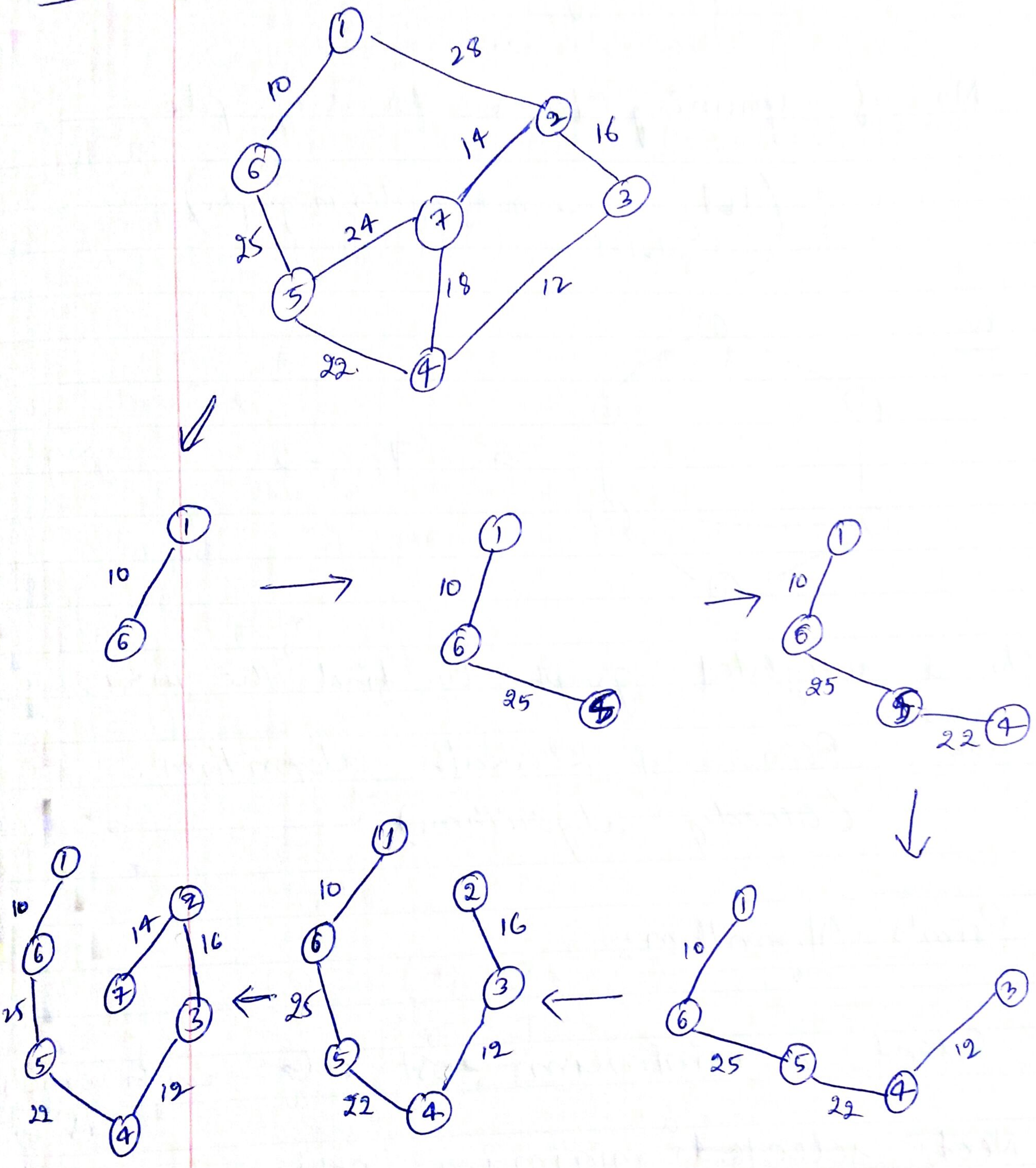
$\Rightarrow \left({}^{7}C_{5} - 2 \right)$ → '2' cycles in graph.

- In a weighted graph, to find the MST, we use Prim's & Kruskal's algorithm. (Greedy algorithms).

Prim's Algorithm :-

1. Select a minimum cost edge with vertices, V_1 & V_2.
2. Next, select a minimum cost edge which is connected to either 'V_1' (or) 'V_2'.

Example:

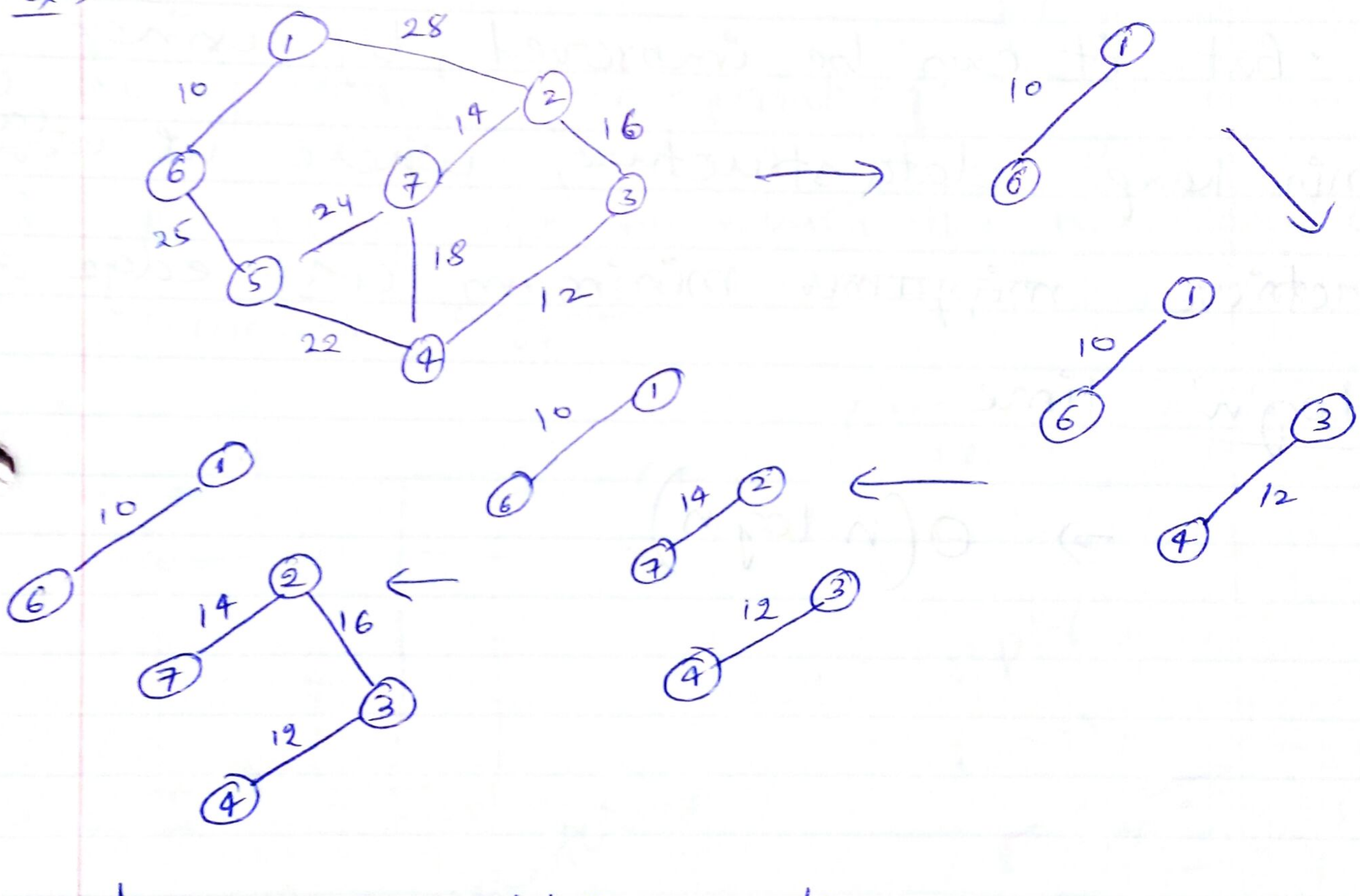


- For non-connected graphs, we cannot find MST.

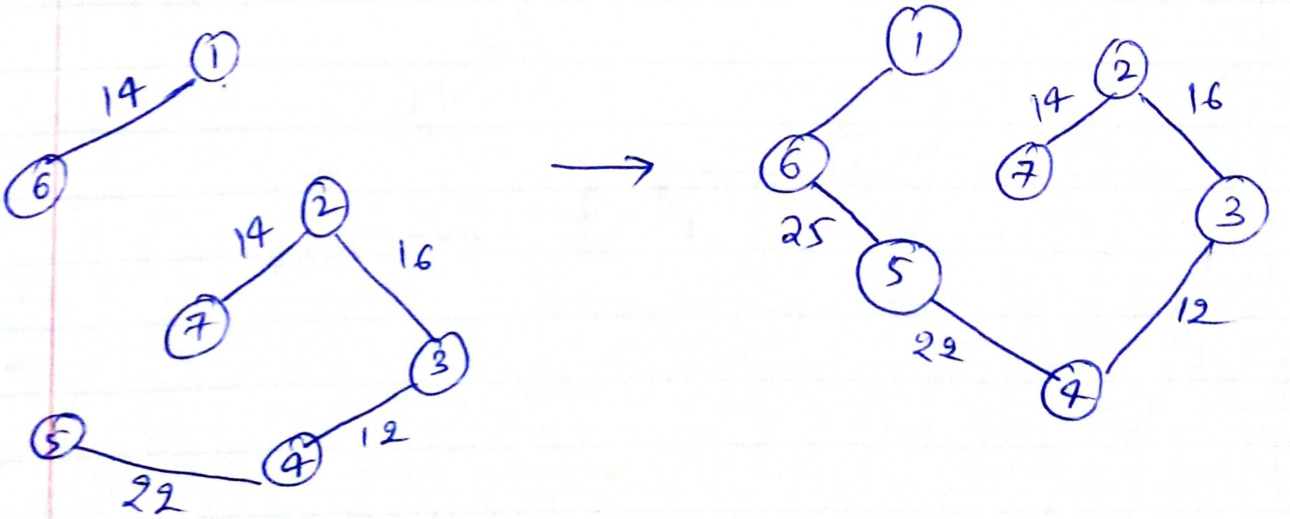
Kruskal's Algorithm :-

- In this algorithm, we always select a minimum cost edge. but only if it does not form a cycle.

Ex :-



edge b/w '4' & '7' forms a cycle
so, discard it



Time complexity :-

- In kruskal, we have to select $|V|-1$, edges out of $|E|$ edges. So, time taken is $O(|E|\,|V|)$. $\Rightarrow O(n^2)$.

- But it can be improved by using "min heap" data structure, where we can retrieve ~~minimum~~ minimum cost edge in "log n" time.

$$\Rightarrow \Theta(n \log n).$$