Module 4: (11 hrs)

Complexity Theory: Class P and NP, Polynomial time reductions, Class NP Hard and NP-Complete Example problems - Vertex cover problem, Clique problem

Network Flows: Flow networks and Network flow, Max-Flow Min Cut Theorem, Ford Fulkerson method, Bipartite matching.

Polynomial Time Algorithms.

The algorithms whose worst case running time is $O(n^k)$ on input size n and constant k, are called polynomial time algorithms.

Deterministic algorithms and Non-deterministic algorithms

Algorithms has the property that the result of every operation is uniquely defined. We allow algorithms to contain operations whose outcomes are not uniquely defined but are limited to specified set of possibility. Such operation is allowed to choose any one of the outcomes subject to a termination condition to be defined later. These leads the concept of a non-deterministic algorithm.

for eg:- Choice (s) are differently choosen one of the element of set S.

Failure ()- Signals an unsuccessful completion

Success ()- Signal a successful completion.

A non-deterministic algorithm terminate unsuccessfully if and only if their exist no set of choices leading to a success signal.

Deterministic and Non-deterministic Algorithm for Search

Deterministic.

```
Algorithm Search (A,n,x)
{
  for i:=1 to n do
  {
    if A[i]:=x then print i;
    return;
    else
      print 0;
    return;
  }
}
```

Non- deterministic

```
j:= choice (1,n);
if A[i] := x then
{
    write j;
    Success ();
}
else
{
    write 0;
    failure();
}
```

Decision and Optimisation Algorithms

Any problem for which the answer is either zero or one (True or false) is called a decision problem. An algorithm for a decision problem is termed as a decision algorithm. An optimization algorithm is used to solve an optimization problem. Any problems that involve the identification of an optimal value of a cost function is known as an optimization problem.

P and NP Classes.

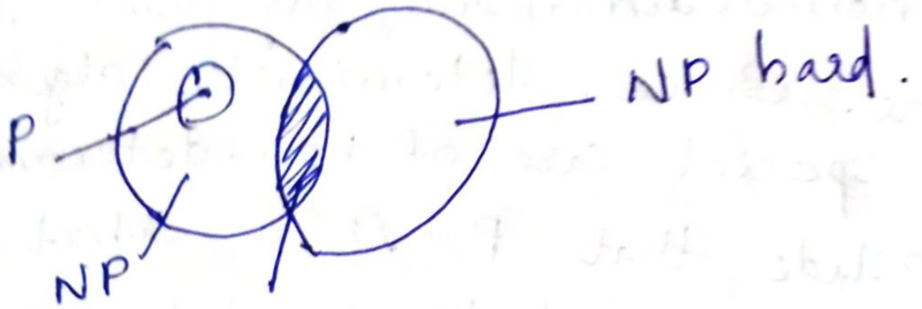
P is the set of all decision problems solvable by deterministic algorithms in polynomial time. NP is the set of all decision problems solvable by non-deterministic algorithms in polynomial time. Since, deterministic algorithms are just a special case of non-deterministic ones, we conclude that P is a subset of NP ($P \subseteq NP$). Commonly believed relationship between P and NP class is

NP hard and NP complete classes.

Let L1 and L2 be problems. Problem L1 reduce to L2 ($L_1 \propto L_2$) if and only if there is a way to solve L1 by a deterministic algorithm that solve L2 in polynomial time.

A problem L is NP-hard iff Satisfiability reduce to L. A problem L is NP complete iff L is NP hard and $L \in NP$ class.

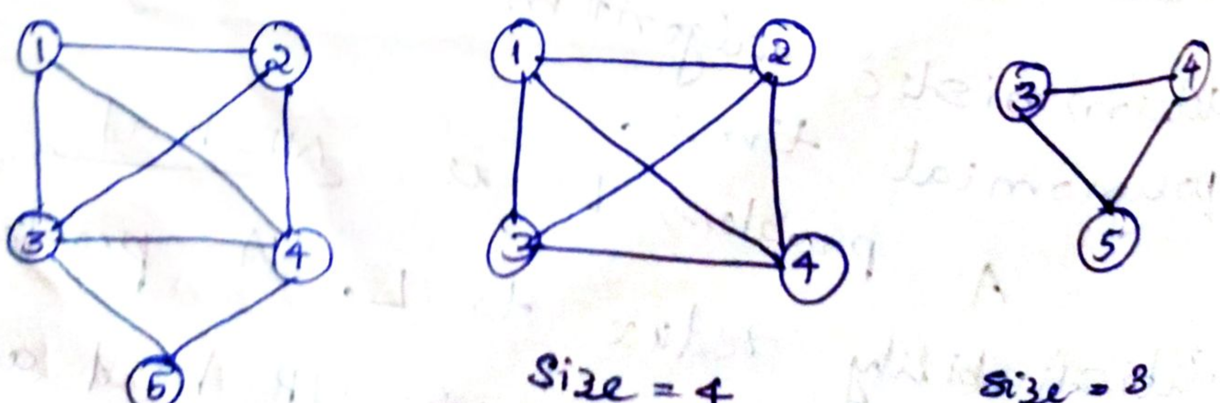
The relation among P, NP, NP hard & NP Complete



Two problems L1 and L2 are said to be polynomial equivalent iff $L1 \alpha L2$ and $L2 \alpha L1$.

Clique Decision Problem (CDP)

A complete graph is a graph in which all the vertices should be connected to each others. ie, there is an edge between every vertices. A maximum complete subgraph of graph is called a clique.



A maximal complete subgraph of a graph G is a clique, the size of the clique is the number of vertices in it. Here the optimization problem is to determine the size of a largest clique in G. The decision problem is to determine whether the G has a clique of the size atleast k for some given k.

CDP is NP-Complete

CDP is NP-complete iff,

① CDP is NP-hard iff satisfiability reduced to CDP.

② CDP belongs to NP class.

According to cooks theorem, it is already proven that satisfiability reduces to CNF satisfiability.

SAT $\propto$ CNF-SAT —— (A)

We have to prove CNF satisfiability reduces to CDP.

CNF-SAT $\propto$ CDP —— (B)

From the above two and according to transitivity

SAT $\propto$ CDP.

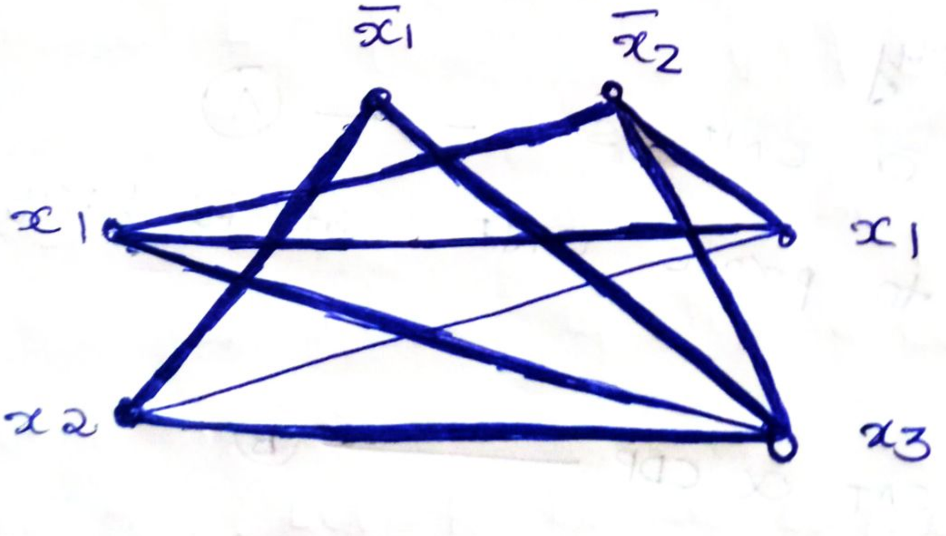
Let $F = \bigwedge_{i=1}^{k} C_i$ be a propositional formula in CNF.

Let x_i be the variable in F. We show how to construct from F a graph G which is equal to $\{V, E\}$ such that G has a clique of size atleast k iff F is satisfiable.

For any F, $G = \{V, E\}$ is defined as follows $E = \{\langle a, i\rangle, \langle b, j\rangle \mid i \neq j \text{ and } b \neq \bar{a}\}$

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

C1 C2 C3

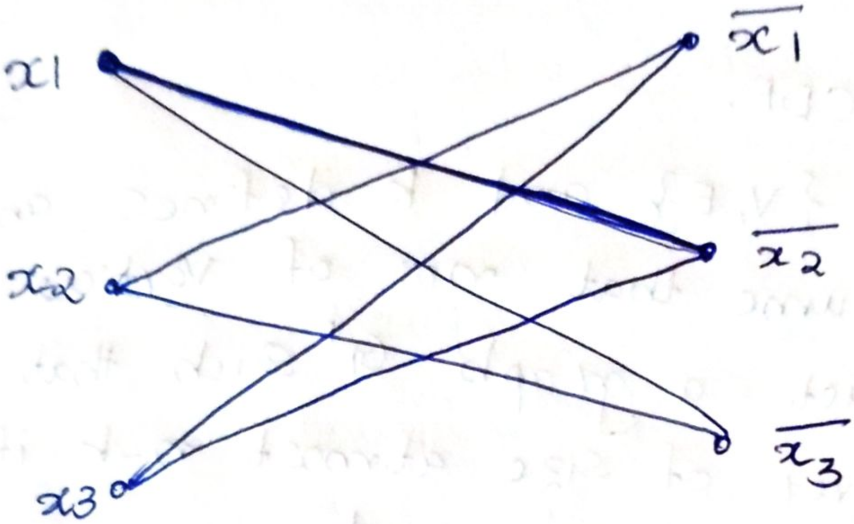


$x_1 = 0$, $x_2 = 1$, and $x_3 = 1$

$(0 \vee 1) \wedge (1 \vee 0) \wedge (0 \vee 1)$

$1 \wedge 1 \wedge 1 = 1$

Eg:- $F = \underbrace{(x_1 \vee x_2 \vee x_3)}_{C1} \wedge \underbrace{(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)}_{C2}$



Given a graph and a value k, we will transform CNF formula into clique problems such that the formula can be satisfied iff the decision problem for the clique is true.

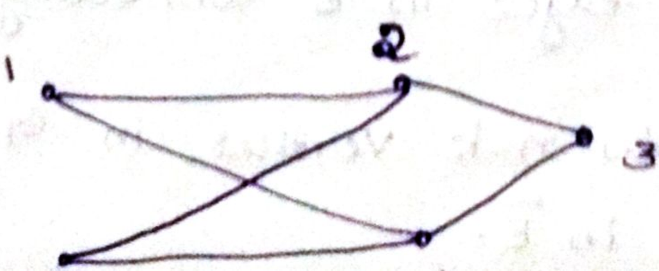
Node Cover Decision Problem (NCDP)

A Set $S \subseteq V$ is a node cover for a graph $G = (V, E)$ iff all edges in E are incident to at least one vertex in S. The size of the cover is the number of vertices in S.

node cover of

$S = \{2, 4\}$ size 2

$S = \{1, 3, 5\}$ size 3.



In the NCDP, we are given a graph G and an integer k. We are required to determine whether G has a Node Cover of size atmost k.

CDP α NCDP.

Let $G = \{V, E\}$ and k defines an instance of CDP assume that no: of vertices is n. We construct a graph G' such that G' has a node cover of size atmost $n-k$ iff G has a clique of size atmost k.

Graph G' is given by

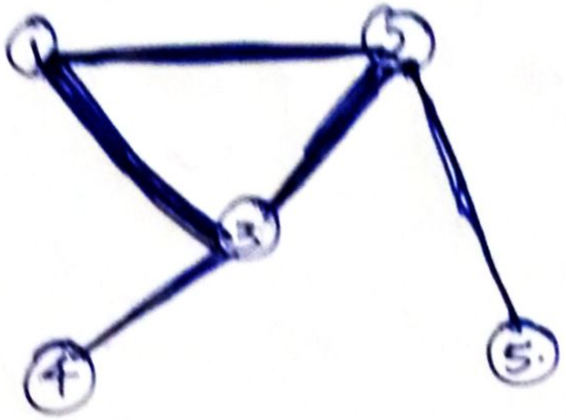
$$G' = (V, \bar{E})$$

$$\text{where } \bar{E} = \{ (u,v) \mid u \in V,\ v \in V \text{ and } (u,v) \notin E \}.$$

The set G' is known as complement of G. Now we show that G has a clique of size atleast k iff G' has a node cover of size atmost $n-k$. Let k be any clique in G since there are no edges in $\bar{E}$ connecting vertices in k.

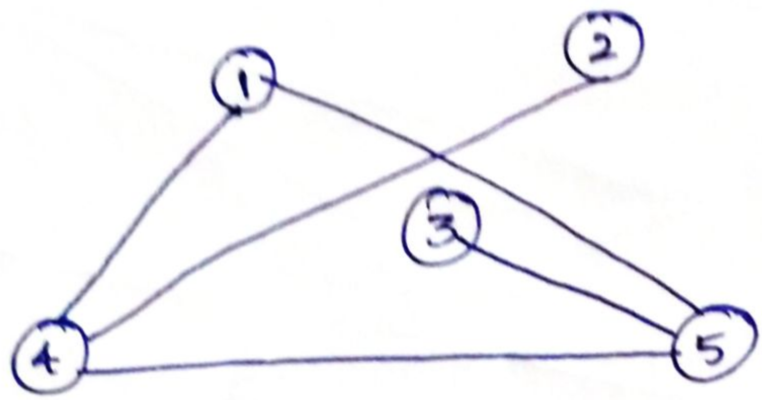
The remain $n-k$ vertices in G must cover all edges in $\bar{E}$.

G =



(1, 3, 5)

G' =



for G,

n = 5, k = 3

for G',

no: Node cover = 5 - 3 = 2

need to find node cover of 2

(4, 5) of size 2.