UNIT-5

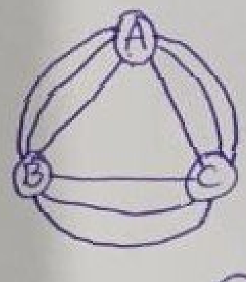
Multi Graphs:

Multigraphs, Bipartite and Planar Graphs, Euler's Theorem, Graph Colouring and Covering, chromatic Number, Spanning Trees; Prim's and Krushal's Algorithms BFS and DFS Spanning Trees.

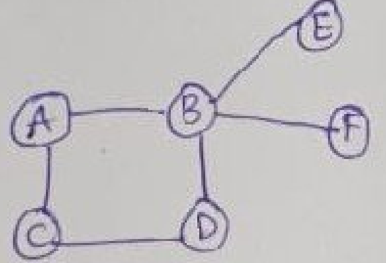
Multi Graphs:

A Graph G(V, E) having no self edges but having parallel edges is called as "Multigraph".

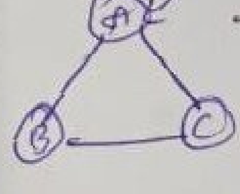
Ex-1:



Ex-2:



Ex-3:



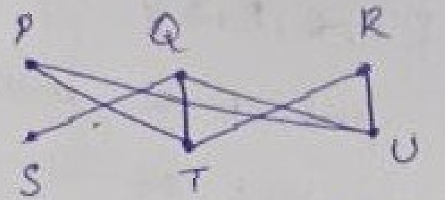
→ it is not multigraph because it is having self-edge.

Bipartite Graph;

A Graph G(V, E) is a bipartite

graph if the vertex V said ~~can~~ can be partitent into two subsets X & Y such that every edges ~~it~~ connect to a vertex in X and a vertex in Y (no edges in the connect~~s~~ either two vertices in X and Y). and it is called "bipartite graph"

Ex-1: Draw $R_{3,3}$ Graph.

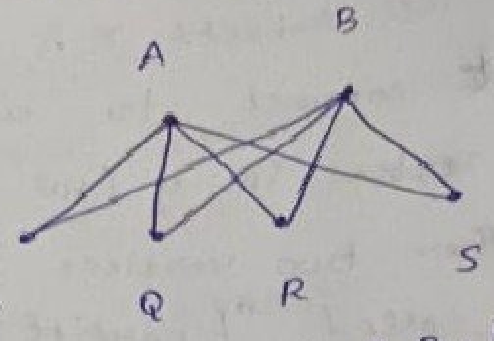


- The vertex $V = \{P, Q, R, S, T, U\}$
- Two subsets are $X = \{P, Q, R\}$
  $Y = \{S, T, U\}$
- The vertex within the same set don't joint. Then we can called as "Bipartite Graph."

Complete Bipartite Graph:

A Graph $G(V, E)$ is complete Bipartite graph if the vertex V can be partitent into two subsets X and Y such that every vertex in X subset is connected to all the vertex in Y subset.

Ex: Draw $K_{2,4}$ Graph.

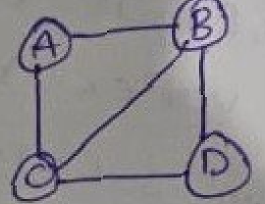


- the vertex $V = \{A, B, P, Q, R, S\}$
- the subset $X = \{A, B\}$
  $Y = \{P, Q, R, S\}$
- Every vertex is subset X connected to all vertexs in subset of Y. then it can called as complete Bipartite graph

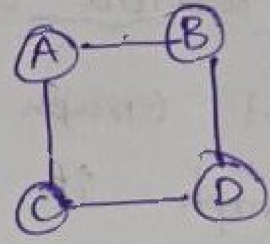
Planar Graph:

A planar graph $G(V, E)$ that we can draw in a plane such that no two edges cross each other.

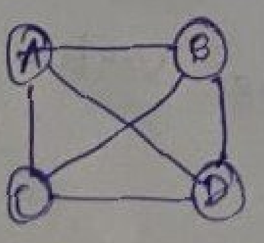
Ex-1:



Ex-2:



Ex-3:

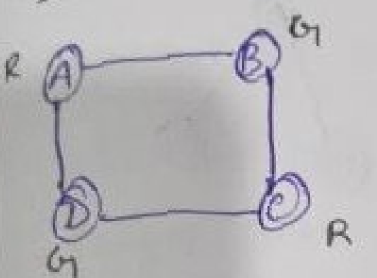


→ It is not a Planar graph.

Graph Colouring:

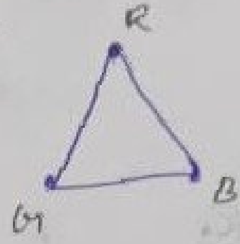
In Graph colouring all the vertices of an undirected graph in such way that no two edges adjacent vertices have the same colour.

Ex-1: R-1 G-2 B-3



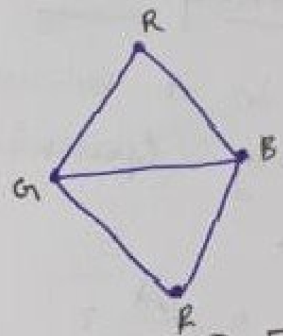
2-colouring

Ex-2:

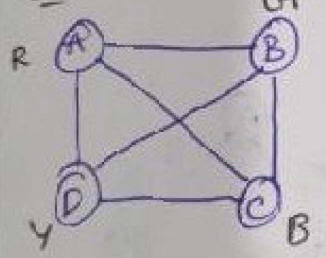


3-colouring

Ex-3:

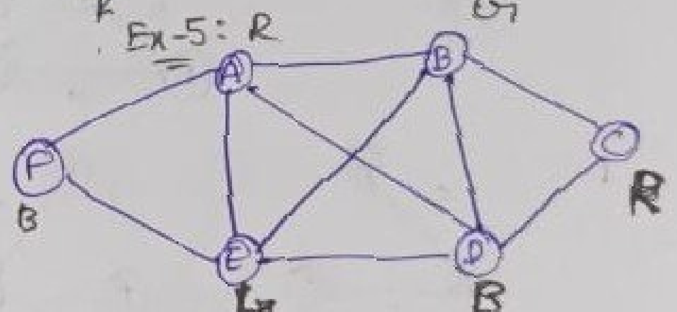


Ex-4:



4-colouring

Ex-5:



4-colouring

Spanning Tree:

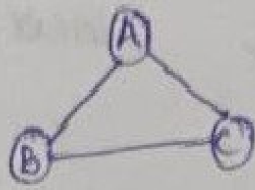
A Spanning tree is a subgraph of given graph G(V, E). Here we can cover of minimum no. of edges it follows some properties:

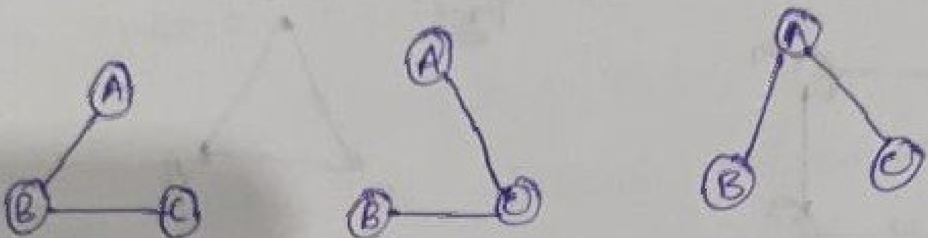
1) The graph does not form cycles.

2) If graph contain N no.of vertices then the spanning tree should contains N-1 edges.

3) Explore all vertices

Ex:

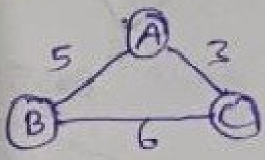


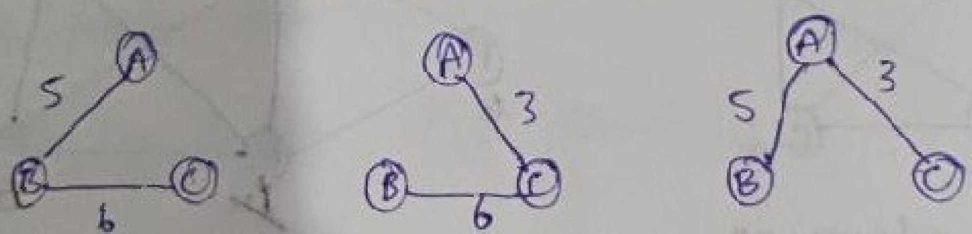


Minimum Spanning tree :

MST is a minimum weight ~~way~~ compare with remaining spanning trees in the same graph.

Ex:





∴ The minimum spanning tree cost is 8.

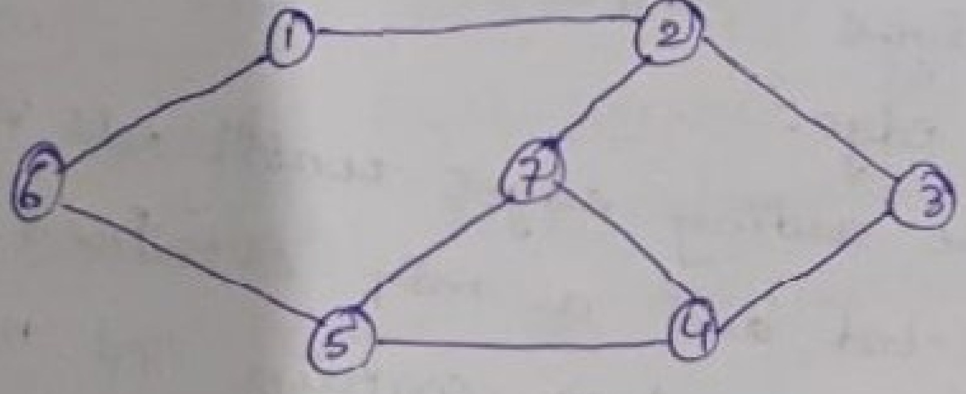
To find minimum spanning tree using two methods. They are

1. prim's Algorithm
2. kruskal's Algorithm

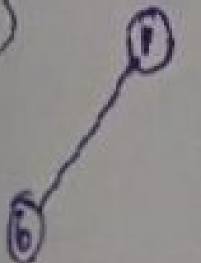
prim's Algorithm:

- Randomly choose any vertex from the given graph.
- selected minimum cost edge for construct spanning tree.
- If cycle form then reject that edge then go to next edge for minimum spanning tree obtained.
- The no. of vertices is N then the edges are N-1.
- Remove all loops and parallel edges.

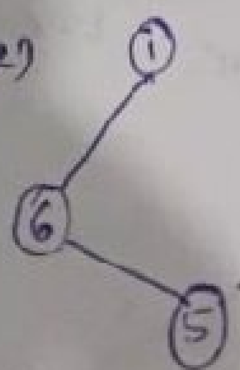
Ex:



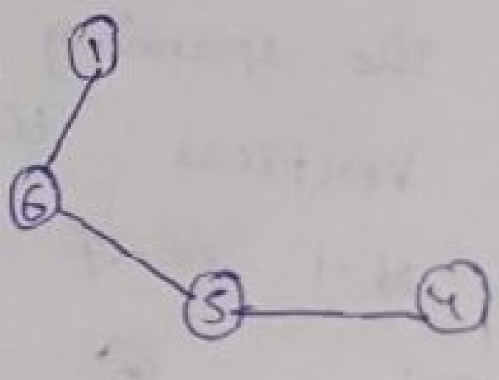
1)



2)

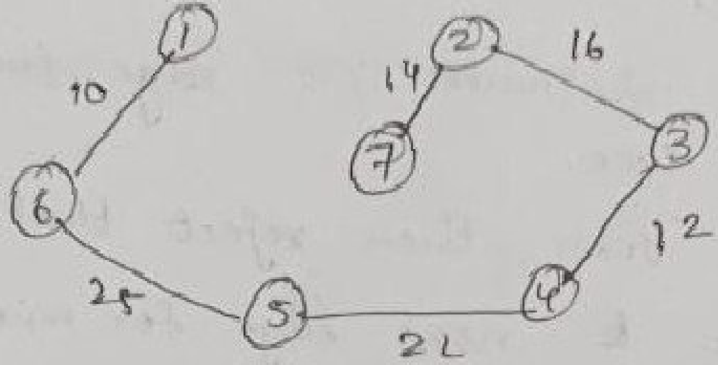


3)



• obtain min spanning tree.

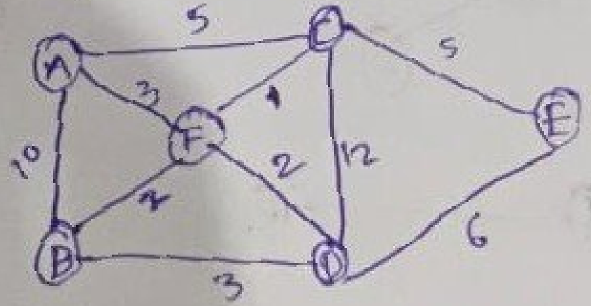
sum of edges weight = 10+25+22+12+16+14
= 99 unit



2) kruskal's Algorithm:

- sort all edges from low weights to highest weight
- Select lowest weight edges then adding an edges creates a cycle then reject that edge.
- If the adding edges until all vertices connected and a MST obtained.
- The spanning tree contain N no.of vertices then the tree contains N-1 no.of edges.

Ex:



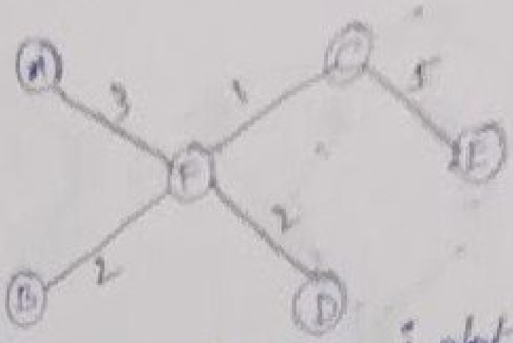
~~decrea~~

C - F = 1 ✓

F - D = 2 ✓

F - B = 3 ✓

F-A = 3
B-D = 3 ✗
A-C = 5 ✗
C-E = 5 ✓
E-D = 6 ✗
A-B = 10 ✗
C-D = 12 ✗



sum of edges weight
= 3+1+2+2+5
= 13 unit.

The spanning tree using two traversal methods:

1. Breath first Search
2. Depth first Search

1. Breath First Search (BFS):

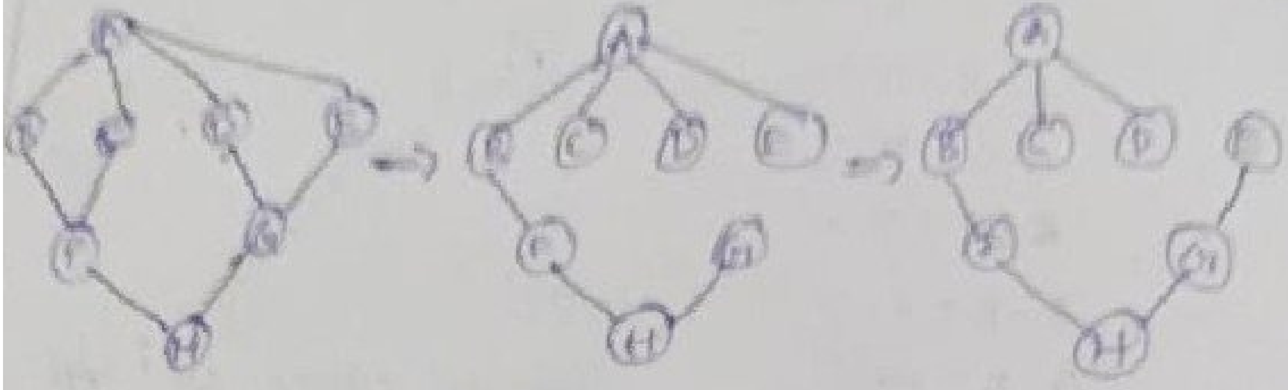
Breath First Search is algorithm for searching tree algorithms. It is used for some of applications.

1. Google maps
2. puzzle.

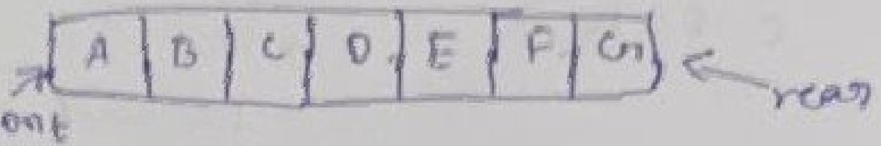
• It is followed some properties.

1. Queue technique (FIFO)

we can start with any vertices in given graph. Explore all vertices connected through selected vertices.



Queue :



Depth First Search (DFS) :

Depth First Search is an algorithm for searching tree algorithms. It is used for some of applications.

- It is followed by some properties.
  1. Stack Technique (FILO)
     From root vertex
  2) pre-order traversal we can traverse from root vertex then left to right vertex.
  3) All vertices & edges and through Selected vertices are traversed recursively.

Ex: