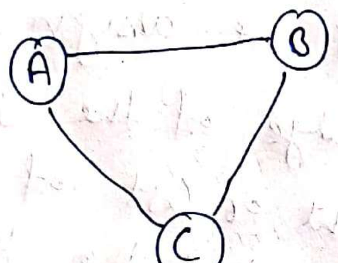
Graph A graph is a non-linear data structure that Consist of two things: (i) a set of points Called vertices or nodes and (ii) a set of edges which Connects the vertices. It is represented by G(V, E) where V is the set of vertices and E is the set of edges. An edge e = (u, v) is a pair of vertices.

eg.



Here V = {A, B, C}
E = {AB, BC, CA}
BA, CB, AC
AB : an edge or link
similary AB, BC, CA.

Graph

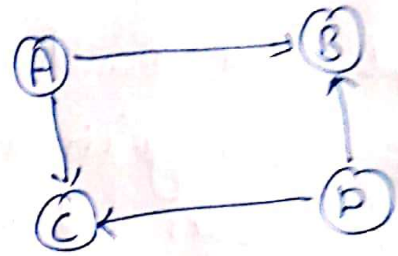
Since ~~there~~ there is no direction from A to B and B to A and similarly for other edges; such type of Graph is Called undirected Graph G.

Vertex : In the above graph, A, B, C are the vertices of the graph i.e. individual data element is Called Vertex.

AB, BA, AC, CA, BC and CB are the edges

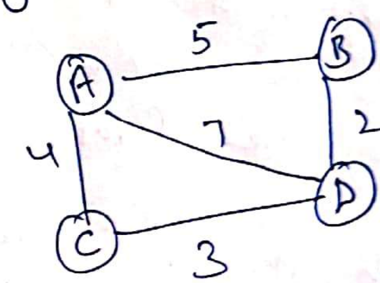
Edge AB, BA, AC, CA, BC and CB are the edges i.e. Connecting link between two vertices. Edge is also Called an arc. These all are undirected edge. These are also Called bidirectional edge.

Directed Edge

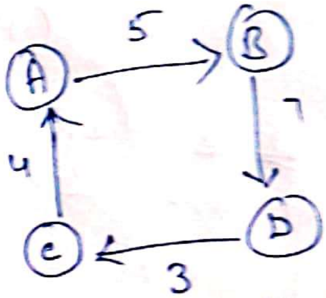


In this graph, we assign a particular direction to each edge. Then that edge is called the directed edge and corresponding graph is called the directed graph.

Weighted Graph: If we assign a non negative number to each edge of the graph which is called the weight or cost of the edge. Such type of graph is called the weighted graph.

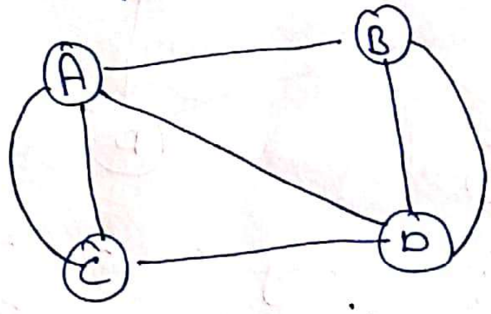


Weighted undirected Graph



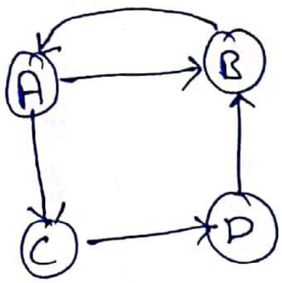
Weighted Directed Graph

Parallel edges or Multiple edges : If in a graph two or more edges joining the same end vertices ~~and~~ is called parallel edges.



Graph (i)

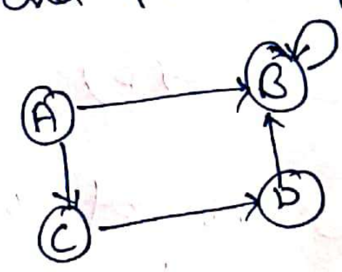
AC and BD are parallel edges in Graph (i)



Graph (ii)

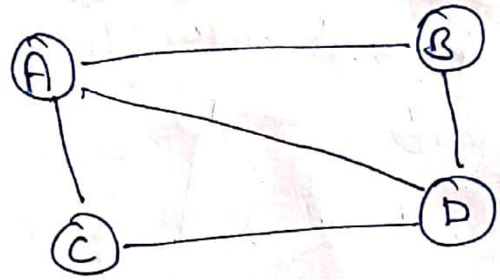
AB is parallel edge in Graph (ii) i.e. directed Graph or digraph

Loop : An edge is called a loop if it has identical end points. eg.



An edge from B to B is a loop in the above G.

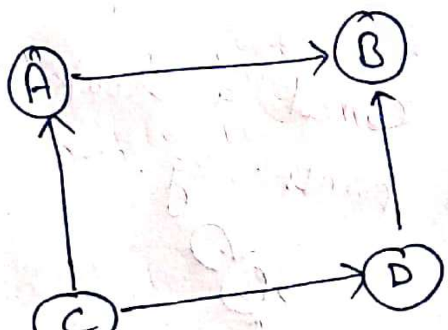
Degree of Edge : total no. of vertices connected to vertex.



deg (A) = 3 ; deg (C) = 2

deg (B) = ~~3~~ 2 ; deg (D) = 3

Indegree total no. of incoming edges connected to a vertex.



indeg (A) = 1

indeg (B) = 2

indeg (C) = 0

indeg (D) = 1

out degree : total no. of outgoing edges connected to a vertex.

eg. for above given directed Graph

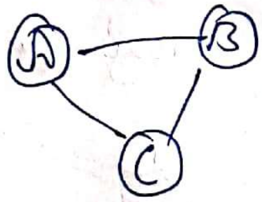
out deg (A) = 1

out deg (B) = 0

out deg (C) = 2

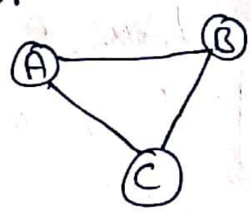
out deg (D) = 1

Simple Graph A graph G is said to be simple if there are no parallel (multiple) edges and no self loops.

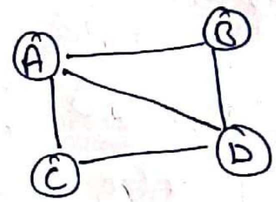


Path A path is a sequence of alternating vertices and edges that start at a vertex and end at a vertex. A path of length 'n' from a node u to a node v is the sequence of (n+1) nodes.

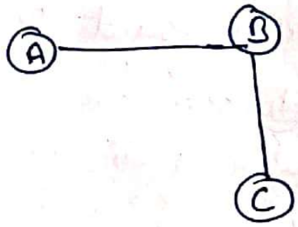
Cycle A cycle is a closed simple path with length 3 or more



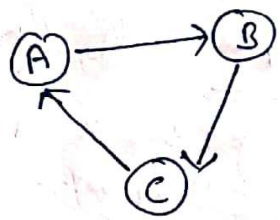
G1



G2



G3

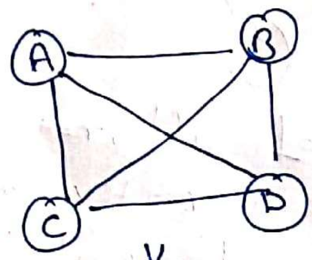


G4

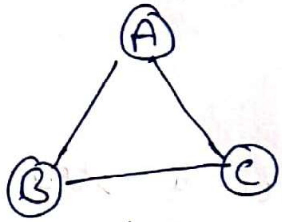
In the above graphs, G_1 and G_2 have cycle. but Graph G does not have any cycle.

In case of directed Graph like in G4 we have to check the direction also.

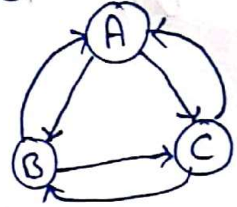
Complete Graph It is a simple graph in which there is exactly one edge between every pair of vertices. eg.



K_4



K_3

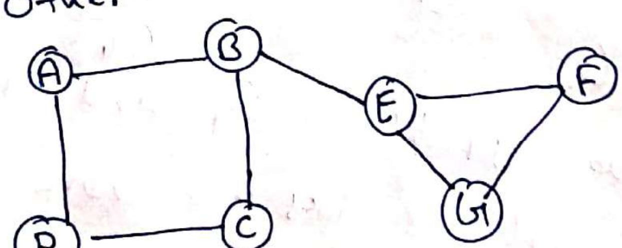


A Complete graph with 'n' vertices is represented by K_n.

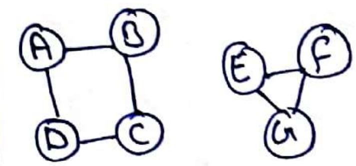
In an undirected Graph with 'n' vertices has exactly $\boxed{\frac{n(n-1)}{2}}$ edges

In directed graph with 'n' vertices has exactly $\boxed{n(n-1)}$ edges

Connected Graph A Graph G is said to be Connected if there exists a path between every pair of vertices. Otherwise the graph is disconnected.

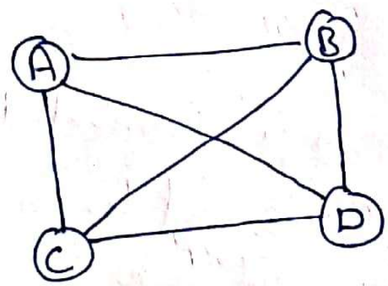


Connected Graph



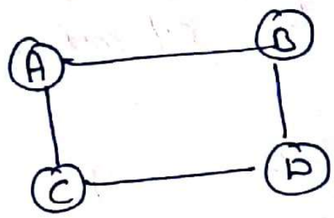
Dis Connected Graph

Regular Graph A graph in which every vertex is of equal deg is called reguler graph. If the degree of every vertex is K then it is called K-regular graph.



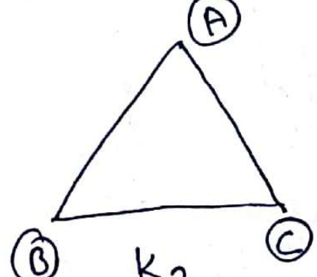
→ deg. of every vertex = 3
So it is Called 3-reguler graph

K_4



→ 2-reguler Graph but not Complete Graph.

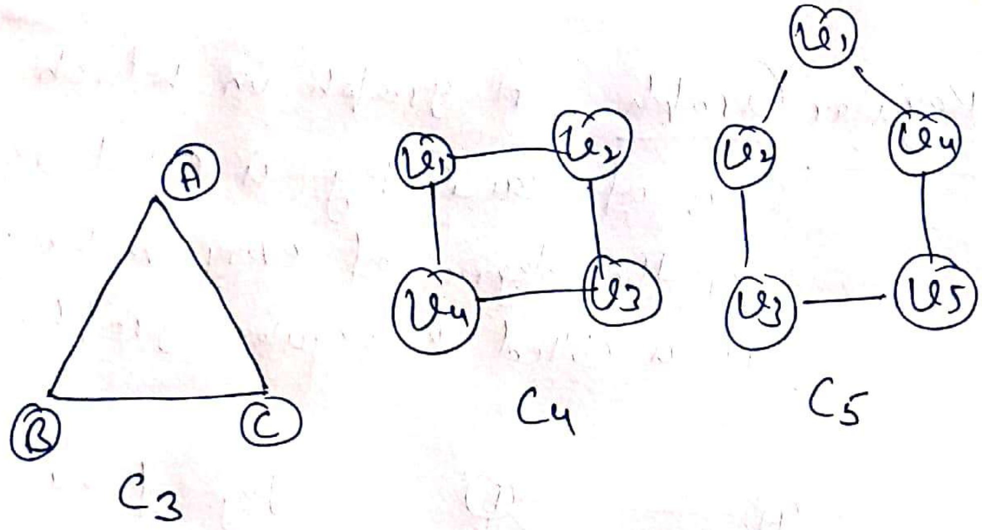
Every Complete Graph is regular but vice versa is not true.



deg. of vertex (A, B, C) = 2.
— Complete as well as regular

K_3

Cycle Graph

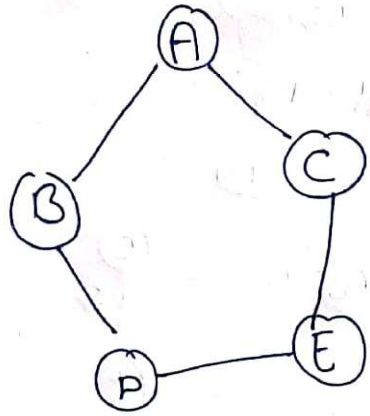


A cycle graph of order n is a Connected graph whose edges form a cycle of length n and it is represented by C_n. Self loops and multiple edges are not considered.

Spanning Tree and Minimum Spanning Tree (MST) of a Graph G(V, E)

[Spanning Tree (ST)]

Consider a graph G(V, E)

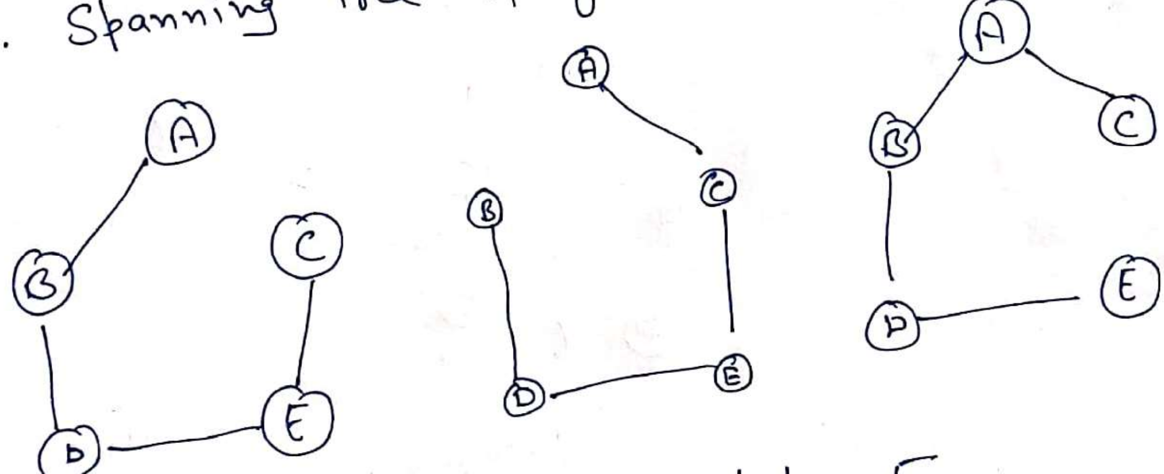


G(V, E)

Then the spanning Tree of a given Graph G (V, E) must Contain all the vertices V of the given G and it should not Contain any cycle. The no. of edges in the spanning tree = |V| - 1

i.e. no. of vertices minus 1.

eg. Spanning tree of given graph are



and so on.

Here |V| = 5

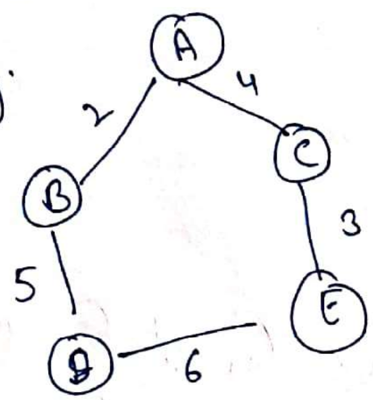
no. of edges in Spanning trees are = 4 i.e. 5-1

Minimum Spanning Tree (MST)

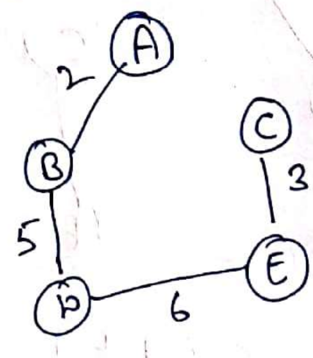
If every edge is assigned some weight (a non-negative number) in the graph, then the spanning tree whose total of edge weight is minimum is called MST.

Cost

eg.



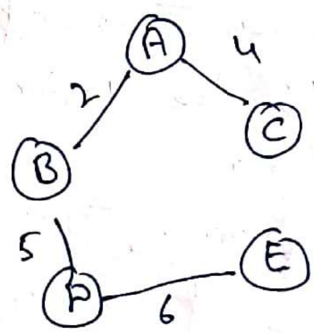
→



(i)

= 16

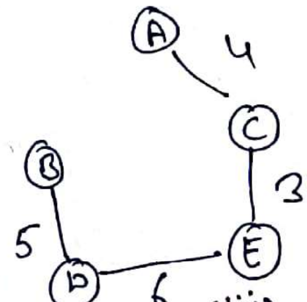
→



(ii)

= 17

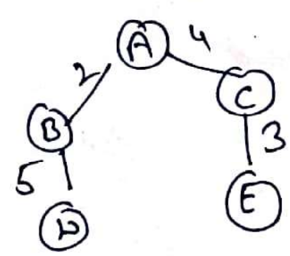
→



(iii)

= 18

→



(iv)

= 14

(iv) ST has minimum Cost, so it is Called the MST.

out of all Spanning Tree (ST)

other Spanning tree has their associated Cost but the Spanning tree formed having Cost 14 is the MST.

Properties of ST

(i) If every edge of the G(V, E) has distinct weight then there will be one and only one MST.

(ii) ST Should not Contain any cycle.

(iii) A Complete undirected G(V, E) having 'n' no. of nodes has n^{n-2} spanning trees.

(iv) Every Connected and undirected Graph has at least one spanning tree.

(v) A Spanning tree should not be disconnected. Removal of any edge from the ST will make it dis Connected.

(vi) If the graph is Complete then removing max (e-n+1) edges, spanning tree Can be formed.

(* Proof of these properties by taking some examples) Verify these properties as an assignment.