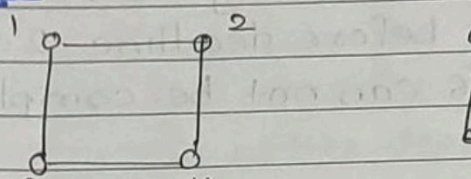
PAGE NO.:
DATE: / /

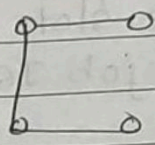
* Spanning Tree :-

A connected subgraph 'S' of Graph (V, E) is said to be spanning if and only if

1. 'S' should contain all vertices of G
2. 'S' should contain (|V| - 1) edges.
3. There should not be cycle present.

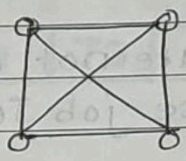
Example





spanning tree

- A single graph can have many different spanning trees.
- If complete graph is given then we can calculate number of spanning trees possible by using a formula i.e. n^{n-2}



K_4 → complete graph.

* Minimum spanning tree :-

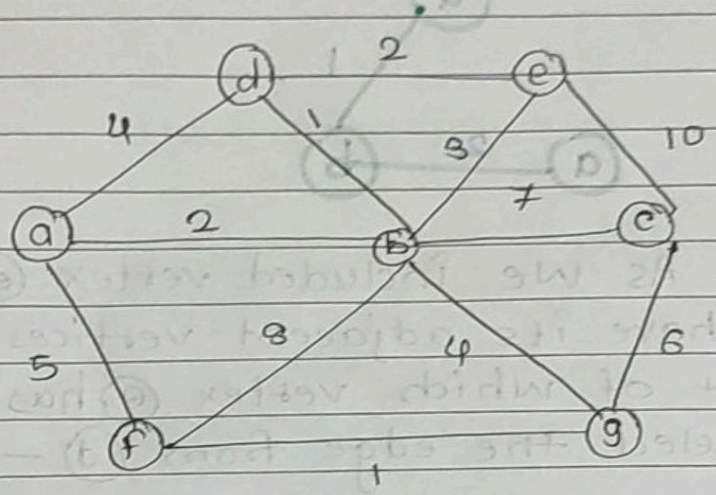
- A minimum spanning tree (MST) is a subset of the edges of connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with minimum possible total edge weight.
- That is it is a spanning tree whose sum of edge weights is as small as possible.

PAGE NO.:
DATE: / /

* Prim's Algorithm :-

- **Step 1 :-** Select any connected vertices with min. weight.
- **Step 2 :-** Select unvisited vertex which is adjacent of visited vertices with minimum weight.
- **Step 3 :-** Repeat 2 until all vertices are visited.

* Example :-

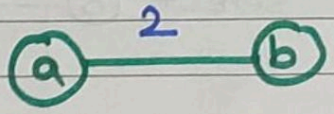


Step 1 :- Choose an arbitary start vertex / any connected vertex

(a)

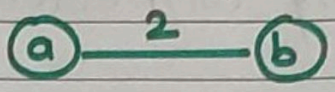
Step 2 :- Select unvisited vertex which is adjacent of visited vertex with min weight.

- In our case (a) has 3 adjacent vertices respectively (d), (f) & (b)
- Therefore we choose a vertex (b) as it has min. weight

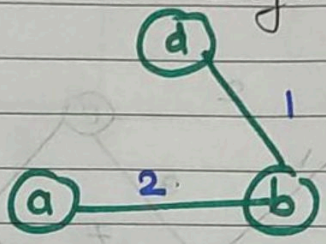


PAGE NO.:
DATE: / /

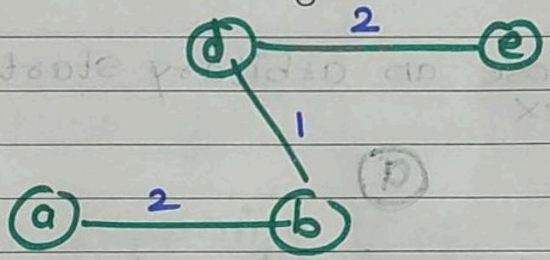
Step 3 :- Repeat step 2 until all vertices are visited



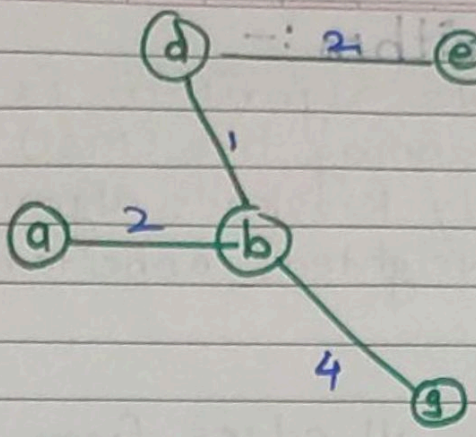
- Now, we have included vertex (b) so we will have its adjacent vertices as (d), (e), (g) & (f), (c)
- out of which vertex (d) has minimum weight so we select the edge from (b) — (d)



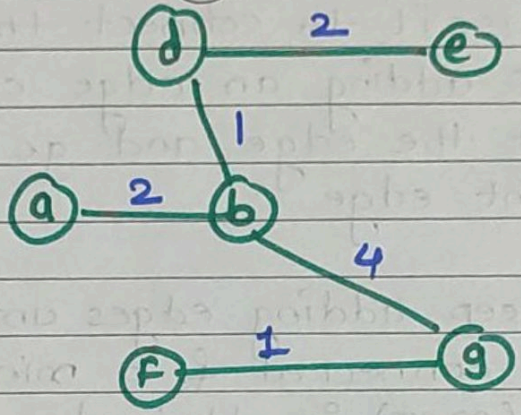
Step 4 :- As we included vertex (d) so we will have its adjacent vertices as (a), & (e)
- out of which vertex (e) has minimum weight so we select the edge from (d) — (e)



Step 5 :- As we have included vertex (e) so we will have its adjacent vertices as (b) & (c)
- out of which vertex (b) has minimum weight but if we include that edge then cycle will be formed so we exclude that edge and select the edge ~~from (e) — (c)~~ which has min weight so we select (b) — (g)

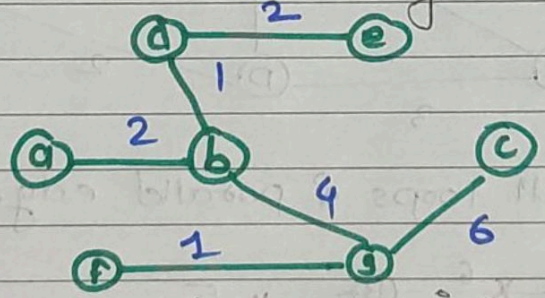


Step 6:- Now we select the adjacent edges of vertex (g) which has min. weight.



Step 7:- Now if we select adjacent edges of vertex (f) then cycle would be formed. So we neglect that adjacent edges.

- Also there is only one vertex (c) is left which is unvisited so we include the vertex (c) with its minimum edge weight.



Step 8:- Now we will add all the weights of edges to get the final cost

$= 2+1+2+1+4+6 = 15$

PAGE NO.:
DATE: / /

* Kruskal's Algorithm :-

- Kruskal's algorithm is used for finding the minimum spanning tree (MST) of a given graph.
- To apply Kruskal's algorithm, the given graph must be weighted, connected & undirected.

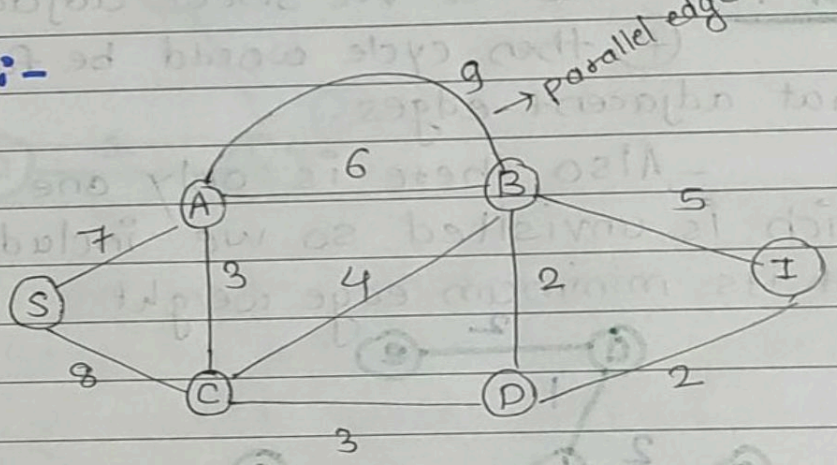
· Algorithm :-

· **Step 1** :- Sort all edges from low weight to high weight.

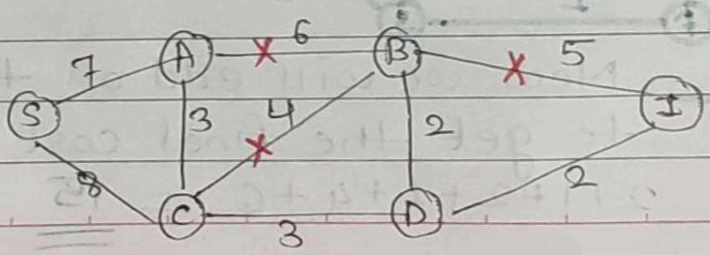
· **Step 2** :- i) Take the edge with the lowest weight and use it to connect the vertices of graph.
ii) If adding an edge creates a cycle, then reject the edge and go for the next least weight edge.

· **Step 3** :- Keep adding edges until all the vertices are connected & a minimum spanning tree (MST) is obtained.

· Example :-



Step 1 :- Remove all loops & parallel edges.

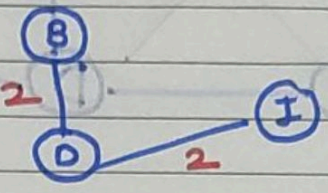


PAGE NO.:
DATE: / /

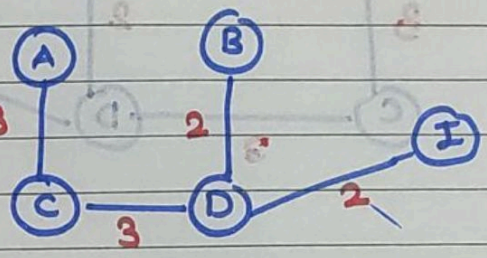
Step 2:- Arrange all edges in their increasing order of weight.

B,D	D,I	A,C	C,D	C,B	B,I	A,B	S,A	S,C
2	2	3	3	4	5	6	7	8

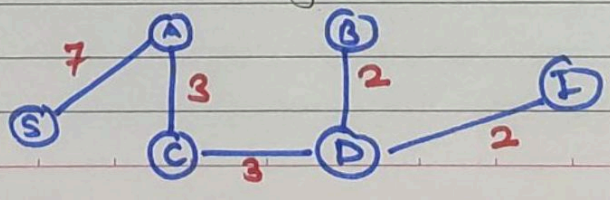
Step 3 :- Add the edge which has least weightage



- Next cost is 3, the edges are AC, & CD, so we select edge (A, C) here & (C, D) here.

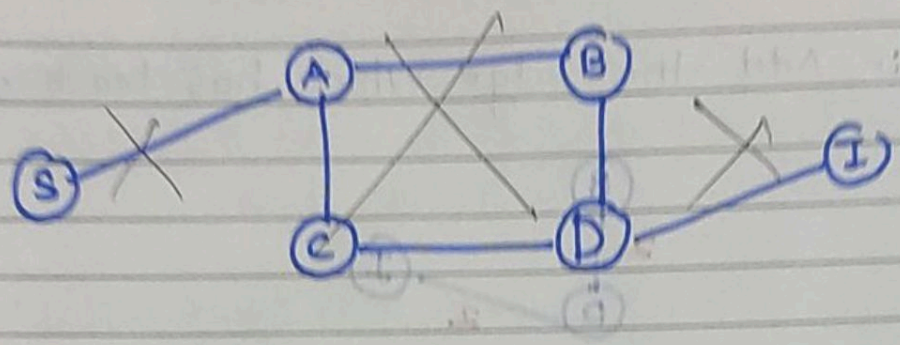


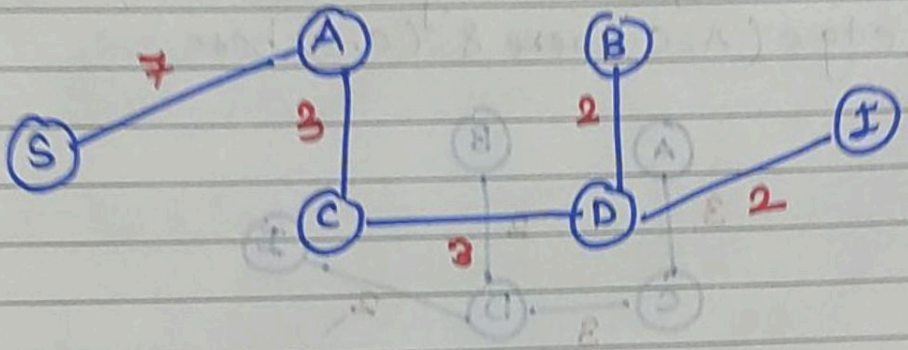
- Next cost is 4, but if we select edge (C,B) it forms a cycle so we discard it.
- Next cost is 5, but if we select edge (B,I) it forms a cycle, so we discard it
- Next cost is 6, but if we select edge (A,B) it again forms a cycle so we discard it.
- Next cost is 7, the edge is (SA) so we select it



- Next cost is 8, but if we select it then it will forms a cycle, so we discard.

Final Minimum Spanning Tree





Weight of MST

= sum of all edges weight

= 7+3+3+2+2

= 17