L-35 Spanning Tree

Wednesday, April 27, 2022 10:02 AM

Spanning Tree :-

Let G be a connected graph. A subgraph T of G is called a spanning tree if

(i) T is a tree.

(ii) T contains all vertices of G.

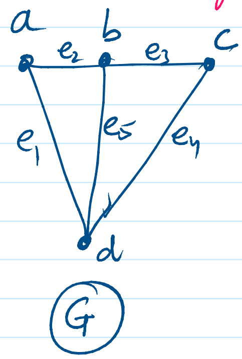
G – graph

T Subgraph of G

How many Edges will be there in T

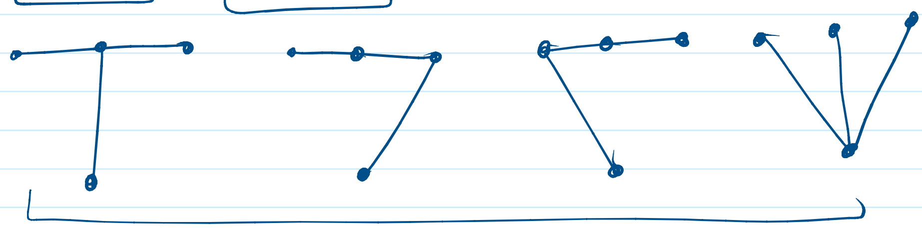
$n-1$

n = No. of Vertices

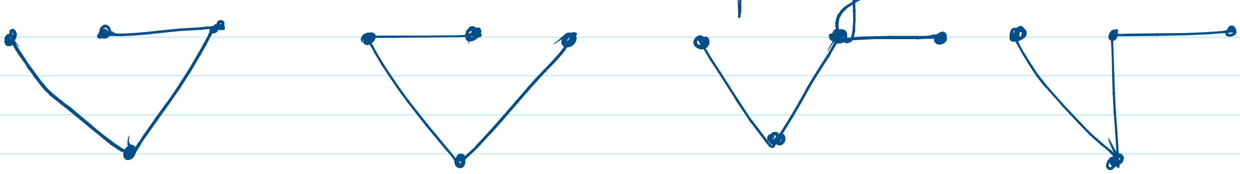


No. of Vertices = 4

$\boxed{n=4}$ $\boxed{n-1=3}$



All are Spanning Trees



Note The spanning Tree of a graph is Not Unique.

Minimal Spanning Tree :→

A minimal spanning tree of a weighted graph is a spanning tree with the Condition that sum of weights of tree is as small as possible.

Maximal Spanning Tree Sum of weight of Tree is as large as possible

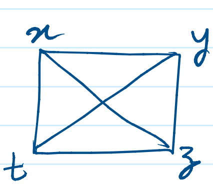
Note :→ The complete graph K_n has n^{n-2} different spanning tree.

K_n n^{n-2} Different S. Tree

How many Sp. Tree are there in the graph K_4

(a) 32 (b) 64 (c) 72 (d) None of these

K_4 $n=4$ $n^{n-2} = (4)^{4-2} = (4)^2 = 16$

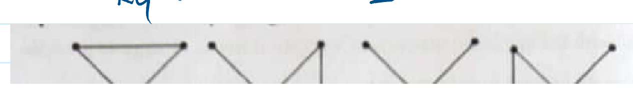


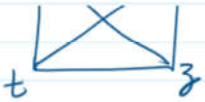
$n=4$

$n-1=3$

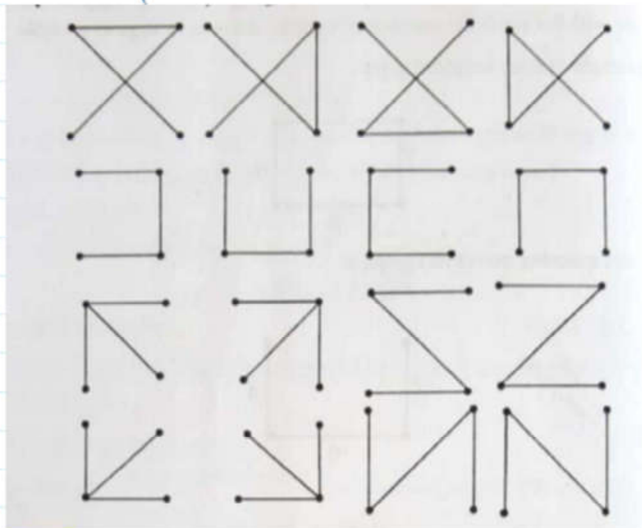
there will be three Edges

Total No. of Edges K_4 $= \frac{4(4-1)}{2} = 6$





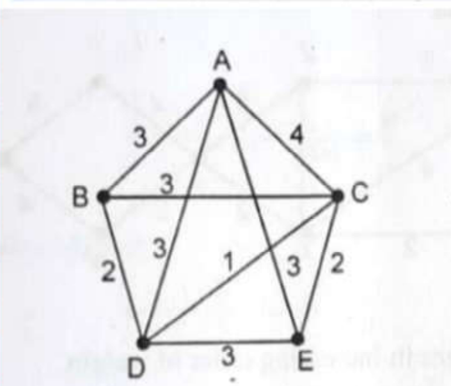
$n-1=3$
there will be three Edges in each Sp. Tree



Art-8. Kruskal's Algorithm to find Minimal Spanning Tree

Let G be the given connected graph with n vertices. Then Kruskal Algorithm to find minimal spanning tree involves following steps :

1. Write all the edges of graph in increasing order of their weight.
2. Select the smallest edge of G.
3. For each successive step select another smallest edge of G which makes no cycle with previously selected edges.
4. Go on repeating step 3 until $n-1$ edges have been selected. The sum of weights of these $n-1$ edges will constitute required minimal spanning tree.

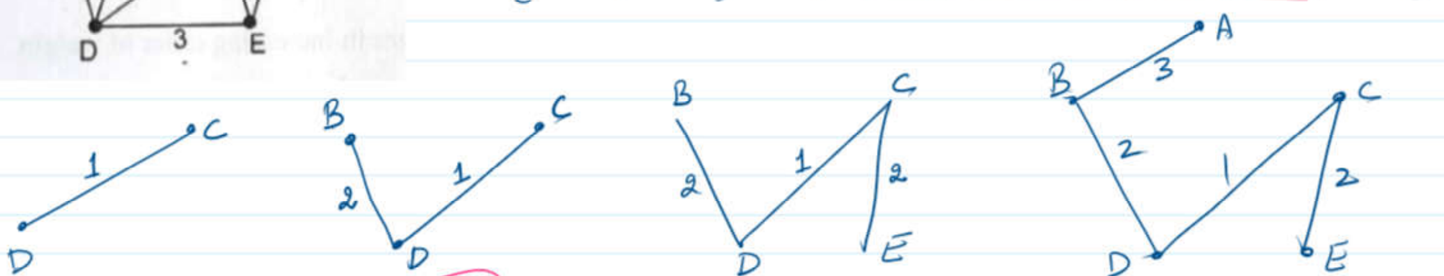


No. of Vertices $= n = 5$

No. of Edges are Need/Required for Sp. Tree $= n-1 = 5-1 = 4$

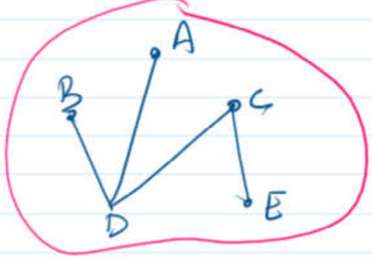
The Edges in Increasing order of their Weight are

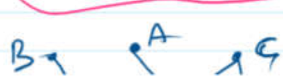
$E = \{ DC, BD, CE, AB, AD, BC, DE, AE, AC \}$

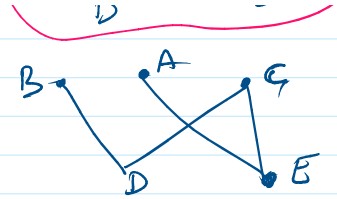


Minimal spanning Tree
$1+2+2+3=8$

or





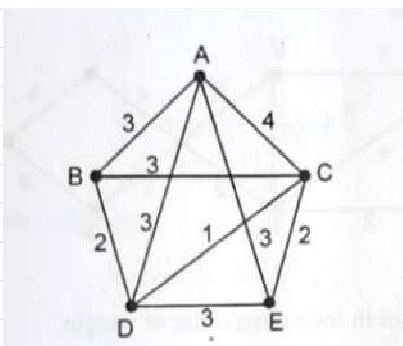


Q2

Art-9. Prim's Algorithm to Find Minimal Spanning Tree

Let G be the given graph with n vertices. Then Prim's algorithm to find minimal spanning tree involves following steps :

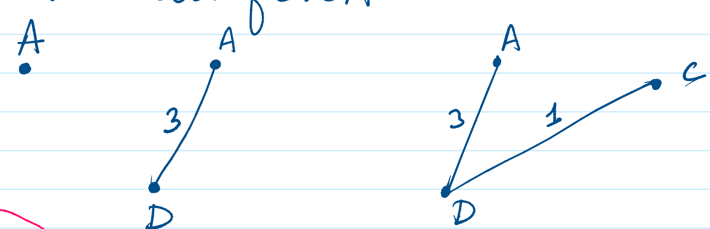
1. Choose any vertex V_1 of G or start from given vertex.
2. Connect V_1 to its nearest neighbour say V_i.
3. Taking (v_1, v_i) as one subgraph, connect this subgraph to its nearest neighbour *i.e.* vertex which is nearest to V_1 or V_i. Let this vertex is V_k. The new vertex must not form a cycle with previous added vertices.
4. Go on repeating step 3 until all n vertices have been connected by $n-1$ edges. The sum of weights of these $n-1$ edges will constitute required minimal spanning tree.

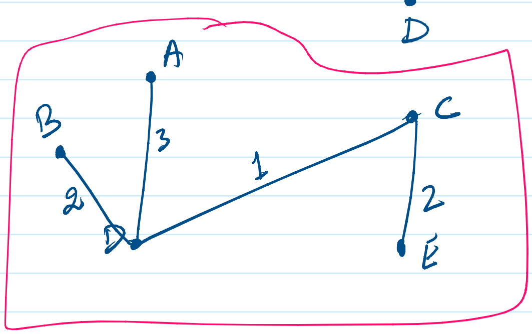


$n = 5$

$n-1 = 5-1 = 4$

We will start from A





Which is the Req. Mini. Sp. Tree

$1 + 2 + 2 + 3 = 8$