

Kruskal's Spanning Tree Algorithm

Kruskal's algorithm is used to find minimum cost spanning tree for a weighted undirected graph using greedy approach.

This algorithm treats the graph as a forest and every node it has as an individual tree. A tree connects to another only and only if, it has the least cost among all available options and does not violate MST (Minimum Spanning Tree) properties.

A **spanning tree** of a graph is just a subgraph that contains all the vertices and is a tree with no cycle. A graph may have many spanning trees.

A spanning tree of a graph with minimum cost is called '**minimum cost spanning tree**'.

Kruskal's Algorithm:

Step 1: Remove all the loops (Any edge that starts and ends at the same vertex is loop)

Step 2: Remove all parallel edges between two vertices except the one with least weight.

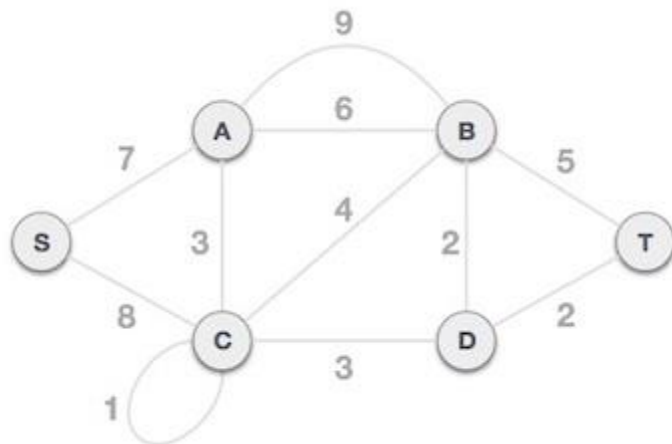
Step 3: Arrange all edges in their increasing order of weight

(for all vertices repeat step 4)

Step 4: Add the edge which has the least weight and with no cycle.

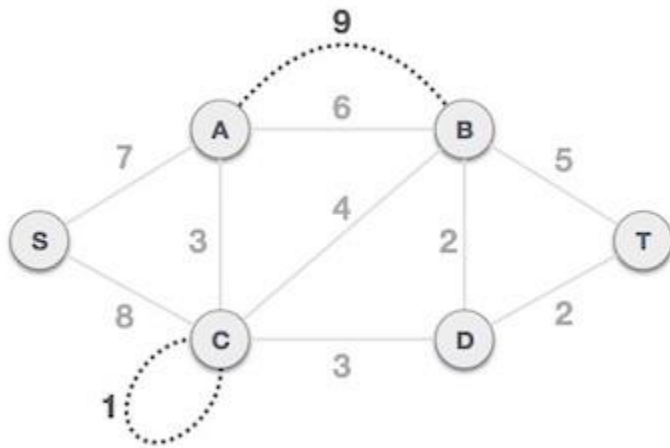
To understand Kruskal's algorithm let us consider the following example –

Find the minimum cost spanning tree from the following graph:

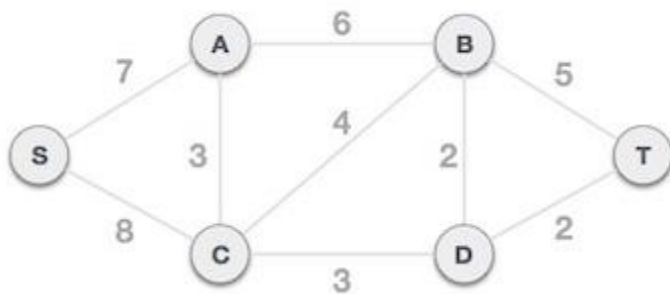


Step 1 and 2 - Remove all loops and Parallel Edges

Remove all loops and parallel edges from the given graph.



In case of parallel edges, keep the one which has the least cost associated and remove all others.



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

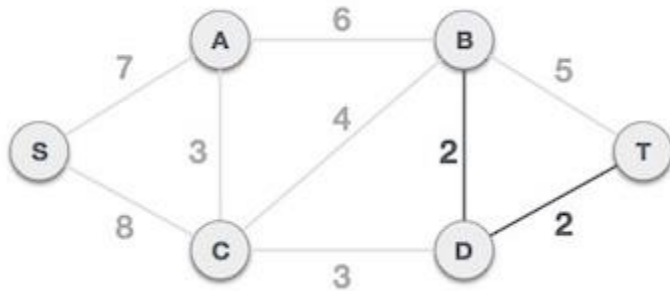
The next step is to create a set of edges and weight, and arrange them in an ascending order of weightage (cost).

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

Step 3 - Add the edge which has the least weight and with no cycle.

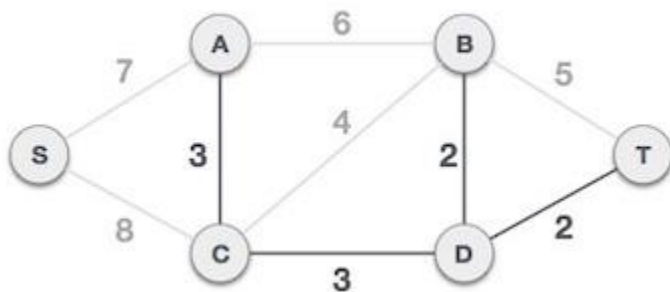
Now we start adding edges to the graph beginning from the one which has the least weight. Throughout, we shall keep checking that the spanning properties remain intact. In case, by

adding one edge, the spanning tree property does not hold then we shall consider not to include the edge in the graph.

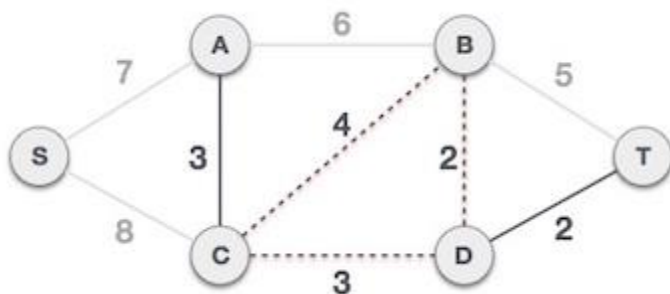


The least cost is 2 and edges involved are B,D and D,T. We add them. Adding them does not violate spanning tree properties, so we continue to our next edge selection.

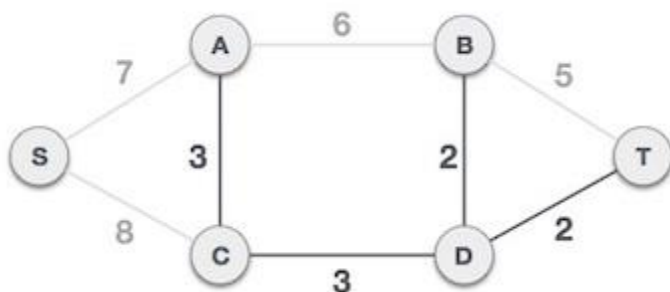
Next cost is 3, and associated edges are A,C and C,D. We add them again –



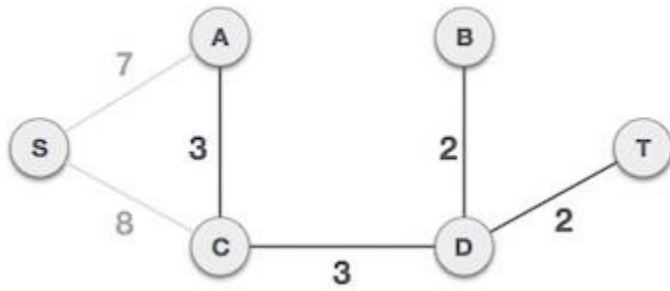
Next cost in the table is 4, and we observe that adding it will create a cycle the graph. –



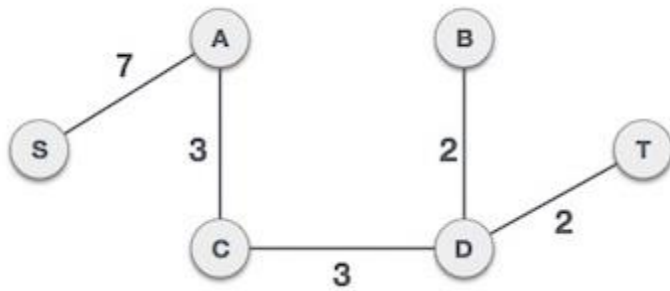
We ignore it. In the process we shall ignore/avoid all edges that create a cycle.



We observe that edges with cost 5 and 6 also create cycle. We ignore them and move on.



Now we are left with only one node to be added. Between the two least cost edges available 7 and 8, we shall add the edge with cost 7.



By adding edge S,A we have included all the nodes of the graph and we now have minimum cost spanning tree.

Now, calculate minimum cost.

Minimum cost of this spanning tree is: $7+3+3+2+2 = 17$

We may find that the output spanning tree of the same graph using two different algorithms is same.