UNIVERSITY INSTITUTE OF COMPUTING

MASTER OF COMPUTER APPLICATIONS

DESIGN AND ANALYSIS OF ALGORITHMS

24CAT-611

UNIT-2

DISCOVER . LEARN . EMPOWER

DESIGN AND ANALYSIS OF ALGORITHMS

Course Outcome

CO Number	Title	Level
CO3	Apply and analyze important algorithmic design paradigms and their applications	Understand
CO4	Implement the major graph algorithms to model engineering problems	Understand

- **Divide and Conquer**: General method, Binary search, Advantages and disadvantages of divide and conquer, Decrease and conquer approach: Topological sort

Topics to be covered

- Minimum Spanning Trees
- Prim’s Algorithm
- Kruskal’s Algorithm

Spanning Tree

- A spanning tree is a subset of Graph G, which has all the vertices covered with minimum possible number of edges. Hence, a spanning tree does not have cycles and it cannot be disconnected..
- By this definition, we can draw a conclusion that every connected and undirected Graph G has at least one spanning tree. A disconnected graph does not have any spanning tree, as it cannot be spanned to all its vertices.
- A complete undirected graph can have maximum $\mathbf{n^{n-2}}$ number of spanning trees, where **n** is the number of nodes. In the above addressed example, **n is 3,** hence 3^{3-2} **= 3** spanning trees are possible.

Minimum Spanning Tree

- *What is Minimum Spanning Tree?*
  Given a connected and undirected graph, a *spanning tree* of that graph is a subgraph that is a tree and connects all the vertices together. A single graph can have many different spanning trees. A *minimum spanning tree (MST)* or minimum weight spanning tree for a weighted, connected and undirected graph is a spanning tree with weight less than or equal to the weight of every other spanning tree. The weight of a spanning tree is the sum of weights given to each edge of the spanning tree.

- *How many edges does a minimum spanning tree has?*
  A minimum spanning tree has (V – 1) edges where V is the number of vertices in the given graph.

Minimum Spanning Tree

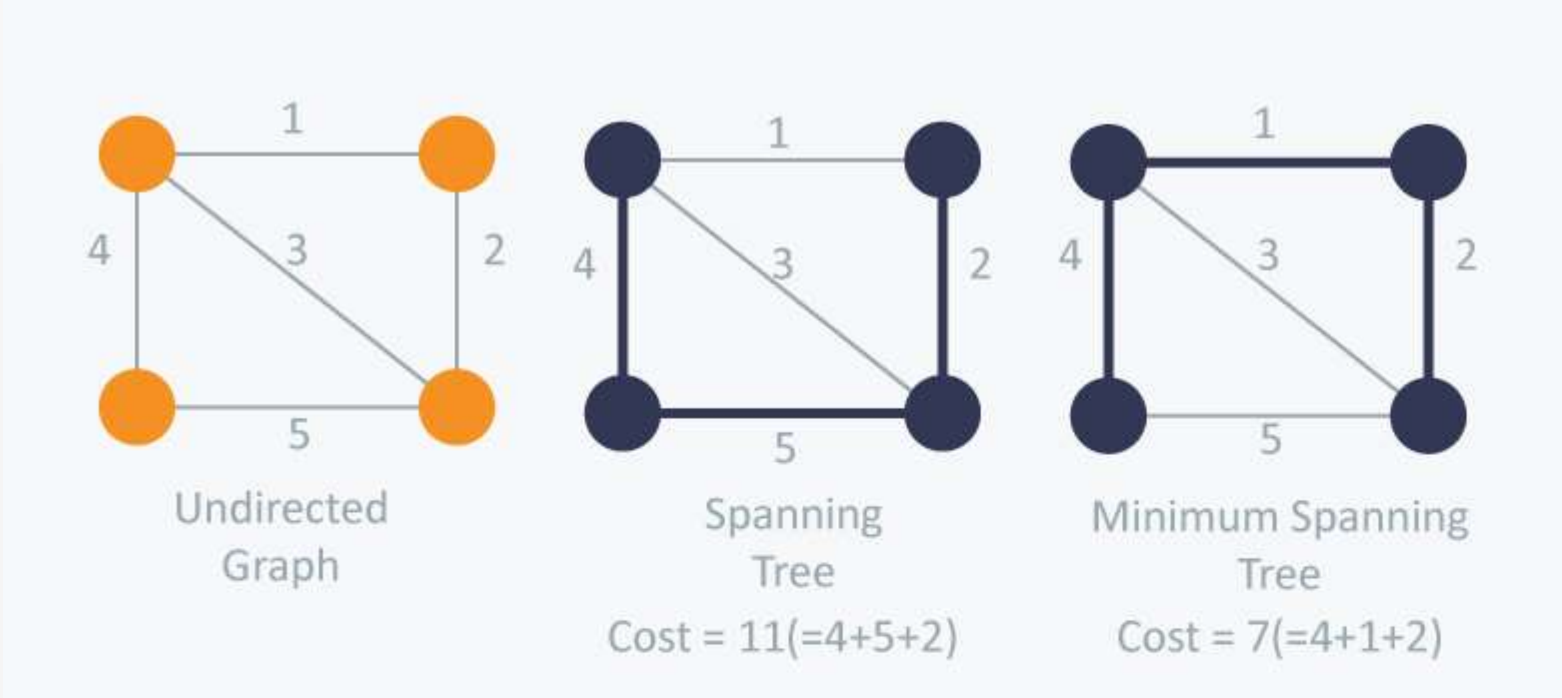

Fig 1: Minimum spanning Tree

Kruskal’s Algorithm

- Kruskal's algorithm to find the minimum cost spanning tree uses the 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 properties.

  The steps for implementing Kruskal's algorithm are as follows:

  1.Sort all the edges from low weight to high

  2.Take the edge with the lowest weight and add it to the spanning tree. If adding the edge created a cycle, then reject this edge.

  3.Keep adding edges until we reach all vertices.

Contd..

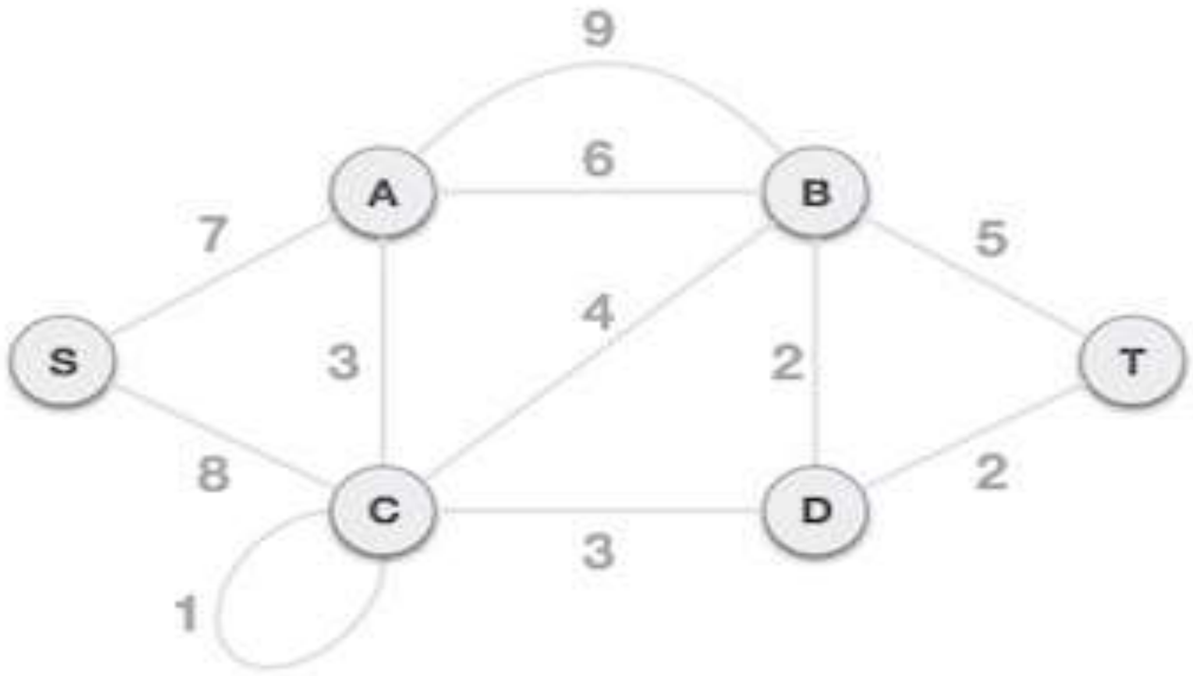

Fig 2: Given Graph

Contd…

- Step 1 - Remove all loops and Parallel Edges

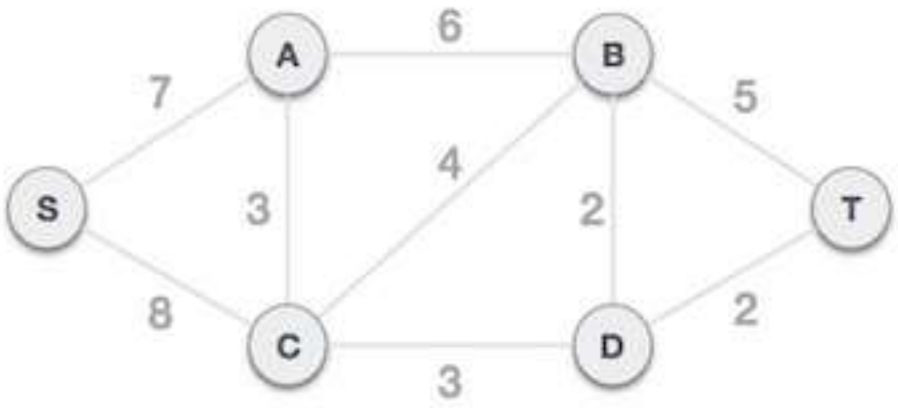

Fig 3: Removed loop and parallel edges

Contd…

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

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

Fig 4: Step 2

Kruskal’s Method

Step 3: Add the edge which has the least weightage

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 −

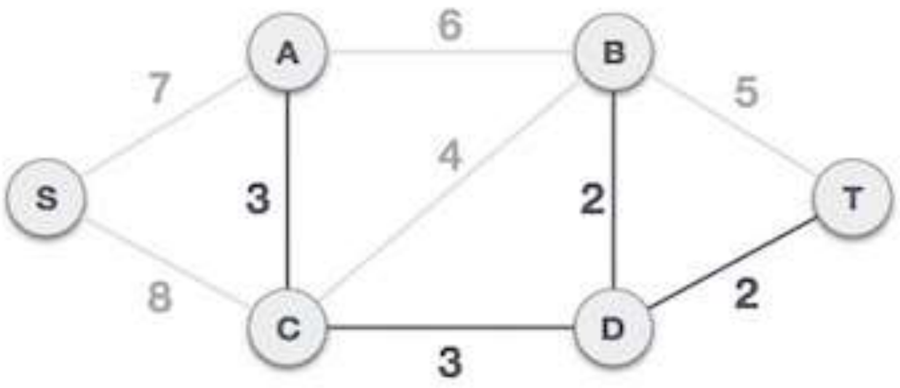

Fig 5: Nest step of Kruskal’s method

Kruskal’s Method

Step 4: Next cost in the table is 4, and we observe that adding it will create a circuit in the graph. −

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

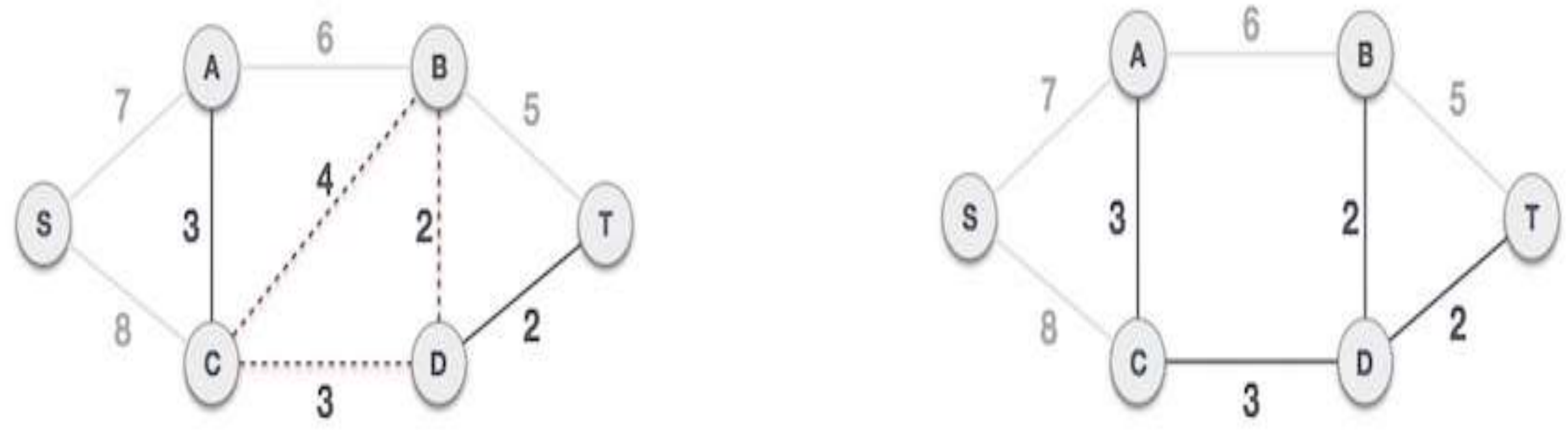

Fig 6: Next step of Kruskal’s method

Contd…

We observe that edges with cost 5 and 6 also create circuits. 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.

Fig 7: Minimum Spanning Tree

References

[1] https://www.tutorialspoint.com/data_structures_algorithms/images/spanning_trees.jpg

[2] https://www.tutorialspoint.com/data_structures_algorithms/images/mst_graph.jpg

[3] https://www.tutorialspoint.com/data_structures_algorithms/images/mst_kruskals_algorithm.jpg

[4] https://www.tutorialspoint.com/data_structures_algorithms/images/mst_kruskals_algorithm.jpg

[5] https://www.tutorialspoint.com/data_structures_algorithms/images/mst_kruskals_algorithm.jpg

[6] https://www.tutorialspoint.com/data_structures_algorithms/images/mst_kruskals_algorithm.jpg

[7] https://www.tutorialspoint.com/data_structures_algorithms/images/mst_kruskals_algorithm.jpg

Books:

1. Introduction to Algorithms by Coreman, Leiserson, Rivest, Stein.
2. Fundamentals of Algorithms by Ellis Horwitz, Sartaj Sahni, Sanguthevar Rajasekaran

THANK YOU