

Modern Education Society's Wadia College of Engineering, Pune

**210256: DATA STRUCTURES and ALGORITHM LABORATORY
(2019 COURSE)**

NAME OF STUDENT:	CLASS:
SEMESTER/YEAR:	ROLL NO:
DATE OF PERFORMANCE:	DATE OF SUBMISSION:
EXAMINED BY:	EXPERIMENT NO: C15

TITLE: PRIM'S ALGORITHM

AIM/PROBLEM STATEMENT: You have a business with several offices; you want to lease phone lines to connect them up with each other; and the phone company charges different amounts of money to connect different pairs of cities. You want a set of lines that connects all your offices with a minimum total cost. Solve the problem by suggesting appropriate data structures.

OBJECTIVES:

1. To understand minimum spanning tree of a Graph
2. To understand how Prim's algorithm works for Minimum Spanning Tree (MST)

OUTCOMES:

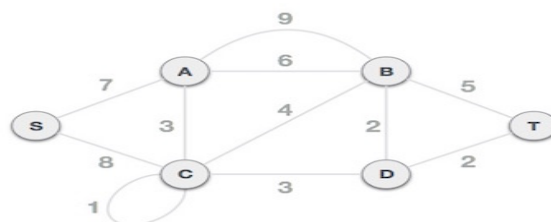
1. To use effective and efficient data structures in solving various Computer Engineering design problems.

PRE-REQUISITE:

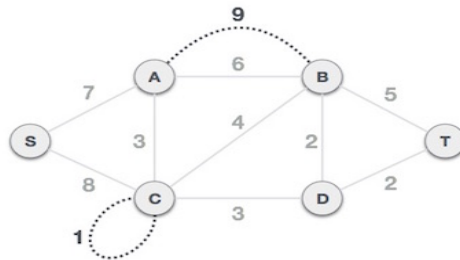
1. Knowledge of C++ programming
2. Knowledge of 2D-Array.

THEORY:

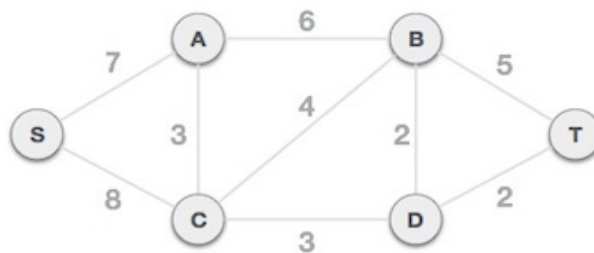
Prim's algorithm to find minimum cost spanning tree (as Kruskal's algorithm) uses the greedy approach. Prim's algorithm shares a similarity with the shortest path first algorithms. Prim's algorithm, in contrast with Kruskal's algorithm, treats the nodes as a single tree and keeps on adding new nodes to the spanning tree from the given graph. To contrast with Kruskal's algorithm and to understand Prim's algorithm better, we shall use the same example –



Step 1 - 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.

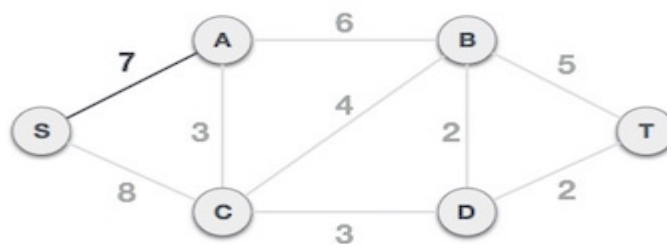


Step 2 - Choose any arbitrary node as root node

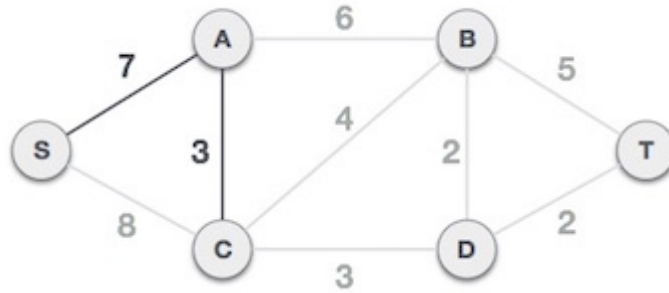
In this case, we choose S node as the root node of Prim's spanning tree. This node is arbitrarily chosen, so any node can be the root node. One may wonder why any video can be a root node. So, the answer is, in the spanning tree all the nodes of a graph are included and because it is connected then there must be at least one edge, which will join it to the rest of the tree.

Step 3 - Check outgoing edges and select the one with less cost

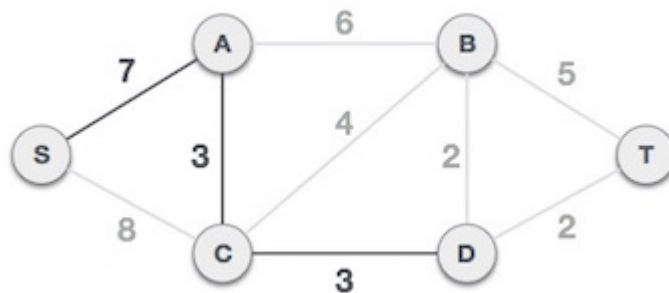
After choosing the root node S, we see that S, A and S, C are two edges with weight 7 and 8, respectively. We choose the edge S, A as it is lesser than the other.



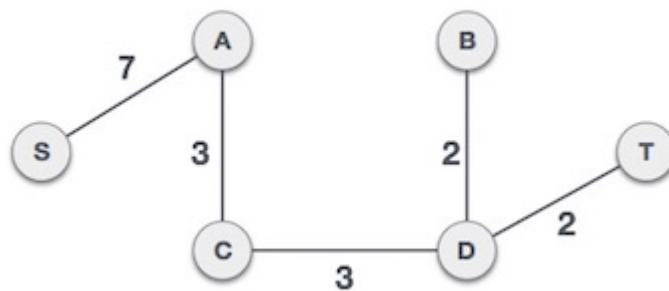
Now, the tree S-7-A is treated as one node and we check for all edges going out from it. We select the one which has the lowest cost and include it in the tree.



After this step, S-7-A-3-C tree is formed. Now we'll again treat it as a node and will check all the edges again. However, we will choose only the least cost edge. In this case, C-3-D is the new edge, which is less than other edges' cost 8, 6, 4, etc.



After adding node **D** to the spanning tree, we now have two edges going out of it having the same cost, i.e., D-2-T and D-2-B. Thus, we can add either one. But the next step will again yield edge 2 as the least cost. Hence, we are showing a spanning tree with both edges included.



The output spanning tree of the same graph using two different algorithms is same.

QUESTIONS:

1. Suppose we have an undirected graph with weights that can be either positive or negative. Do Prim's and Kruskal's algorithm produce an MST for such a graph?
2. Can a graph have more than one spanning tree?
3. State the difference between Prim's and Kruskal's Algorithm.