Assignment no: 3

Prims Algorithm :

The algorithm starts with an empty spanning tree. The idea is to maintain two sets of vertices. The first set contains the vertices already included in the MST, and the other set contains the vertices not yet included. At every step, it considers all the edges that connect the two sets and picks the minimum weight edge from these edges. After picking the edge, it moves the other endpoint of the edge to the set containing MST.

A group of edges that connects two sets of vertices in a graph is called cut in graph theory. So, at every step of Prim's algorithm, find a cut, pick the minimum weight edge from the cut, and include this vertex in MST Set (the set that contains already included vertices).

How does Prim's Algorithm Work?

Step 1: Determine an arbitrary vertex as the starting vertex of the MST.
Step 2: Follow steps 3 to 5 till there are vertices that are not included in the MST (known as fringe vertex).
Step 3: Find edges connecting any tree vertex with the fringe vertices.
Step 4: Find the minimum among these edges.
Step 5: Add the chosen edge to the MST if it does not form any cycle.
Step 6: Return the MST and exit

How to implement Prim's Algorithm?

Follow the given steps to utilize the **Prim's Algorithm** mentioned above for finding MST of a graph:

- Create a set **mstSet** that keeps track of vertices already included in MST.
- Assign a key value to all vertices in the input graph. Initialize all key values as INFINITE. Assign the key value as 0 for the first vertex so that it is picked first.
- While **mstSet** doesn't include all vertices
    - Pick a vertex **u** that is not there in **mstSet** and has a minimum key value.
    - Include **u** in the **mstSet**.
    - Update the key value of all adjacent vertices of **u**. To update the key values, iterate through all adjacent vertices.

- For every adjacent vertex **v**, if the weight of edge **u-v** is less than the previous key value of **v**, update the key value as the weight of **u-v**.

The idea of using key values is to pick the minimum weight edge from the cut. The key values are used only for vertices that are not yet included in MST, the key value for these vertices indicates the minimum weight edges connecting them to the set of vertices included in MST.

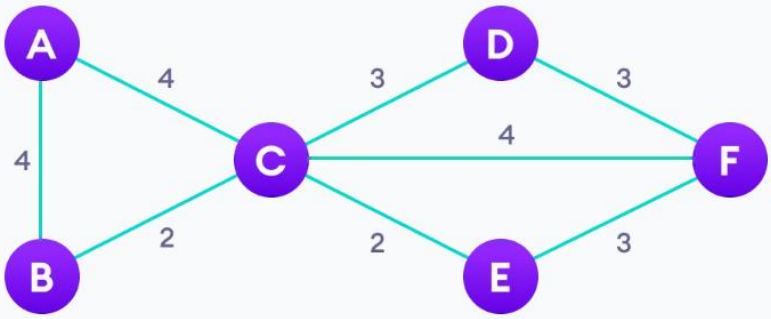
Algorithm

1. Step 1: Select a starting vertex
2. Step 2: Repeat Steps 3 and 4 until there are fringe vertices
3. Step 3: Select an edge 'e' connecting the tree vertex and fringe vertex that has minimum weight
4. Step 4: Add the selected edge and the vertex to the minimum spanning tree T
5. [END OF LOOP]
6. Step 5: EXIT

Time Complexity: $O(V^2)$, If the input graph is represented using an adjacency list, then the time complexity of Prim's algorithm can be reduced to O(E * logV) with the help of a binary heap. In this implementation, we are always considering the spanning tree to start from the root of the graph
Auxiliary Space: O(V)

Time Complexity: O(E*log(E)) where E is the number of edges
Auxiliary Space: O(V^2) where V is the number of vertex

EXAMPLE OF PRIM'S ALGORITHM

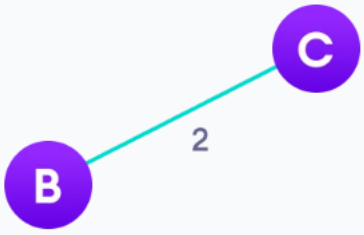


Step: 1

Start with a weighted graph

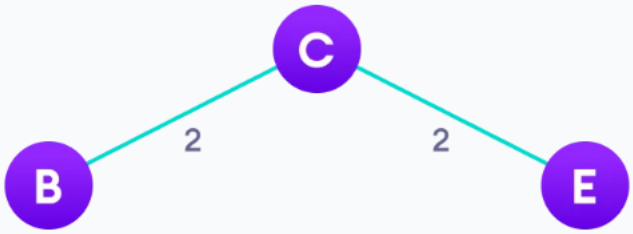
Step: 2

Choose a vertex



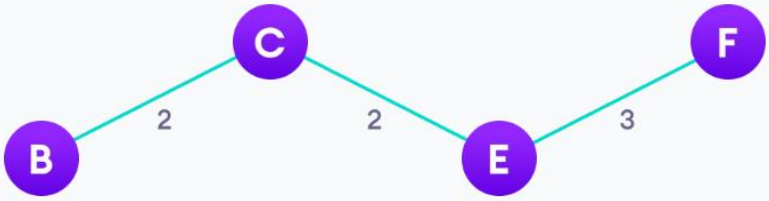
Step: 3

Choose the shortest edge from this vertex and add it



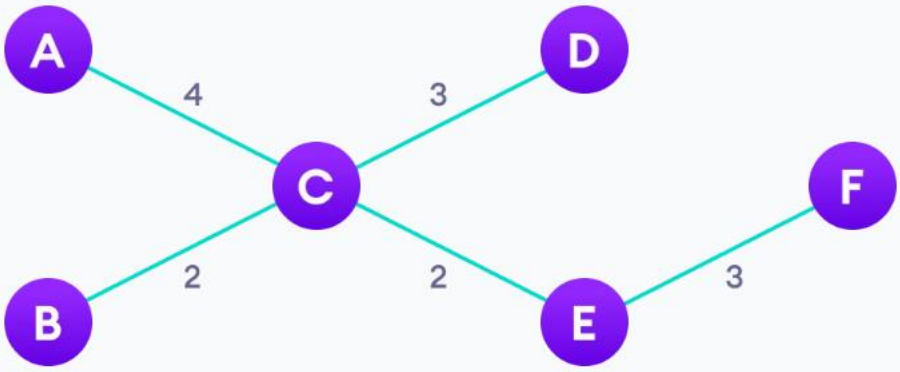
Step: 4

Choose the nearest vertex not yet in the solution



Step: 5

Choose the nearest edge not yet in the solution, if there are multiple choices, choose one at random



Step: 6

Repeat until you have a spanning tree

Advantages:

1. Prim’s algorithm is guaranteed to find the MST in a connected, weighted graph.
2. It has a time complexity of O(E log V) using a binary heap or Fibonacci heap, where E is the number of edges and V is the number of vertices.
3. It is a relatively simple algorithm to understand and implement compared to some other MST algorithms.

Disadvantages:

1. Like Kruskal’s algorithm, Prim’s algorithm can be slow on dense graphs with many edges, as it requires iterating over all edges at least once.
2. Prim’s algorithm relies on a priority queue, which can take up extra memory and slow down the algorithm on very large graphs.
3. The choice of starting node can affect the MST output, which may not be desirable in some applications

The applications of prim's algorithm are -

- Prim's algorithm can be used in network designing.
- It can be used to make network cycles.
- It can also be used to lay down electrical wiring cables.