DATA STRUCTURES AND ALGORITHMS

DR SAMABIA TEHSIN

BS (AI)

Spanning Tree and Minimum Spanning Tree

Spanning tree

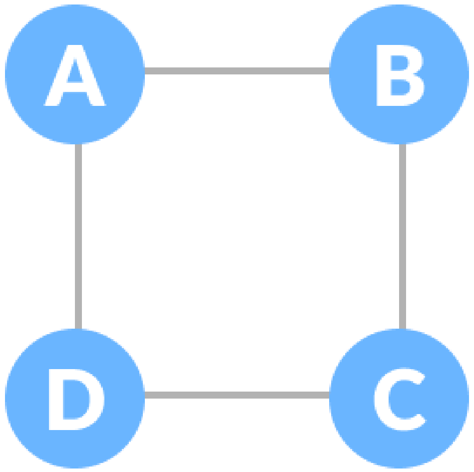
A spanning tree is a sub-graph of an undirected connected graph, which includes all the vertices of the graph with a minimum possible number of edges. If a vertex is missed, then it is not a spanning tree.

Spanning Tree

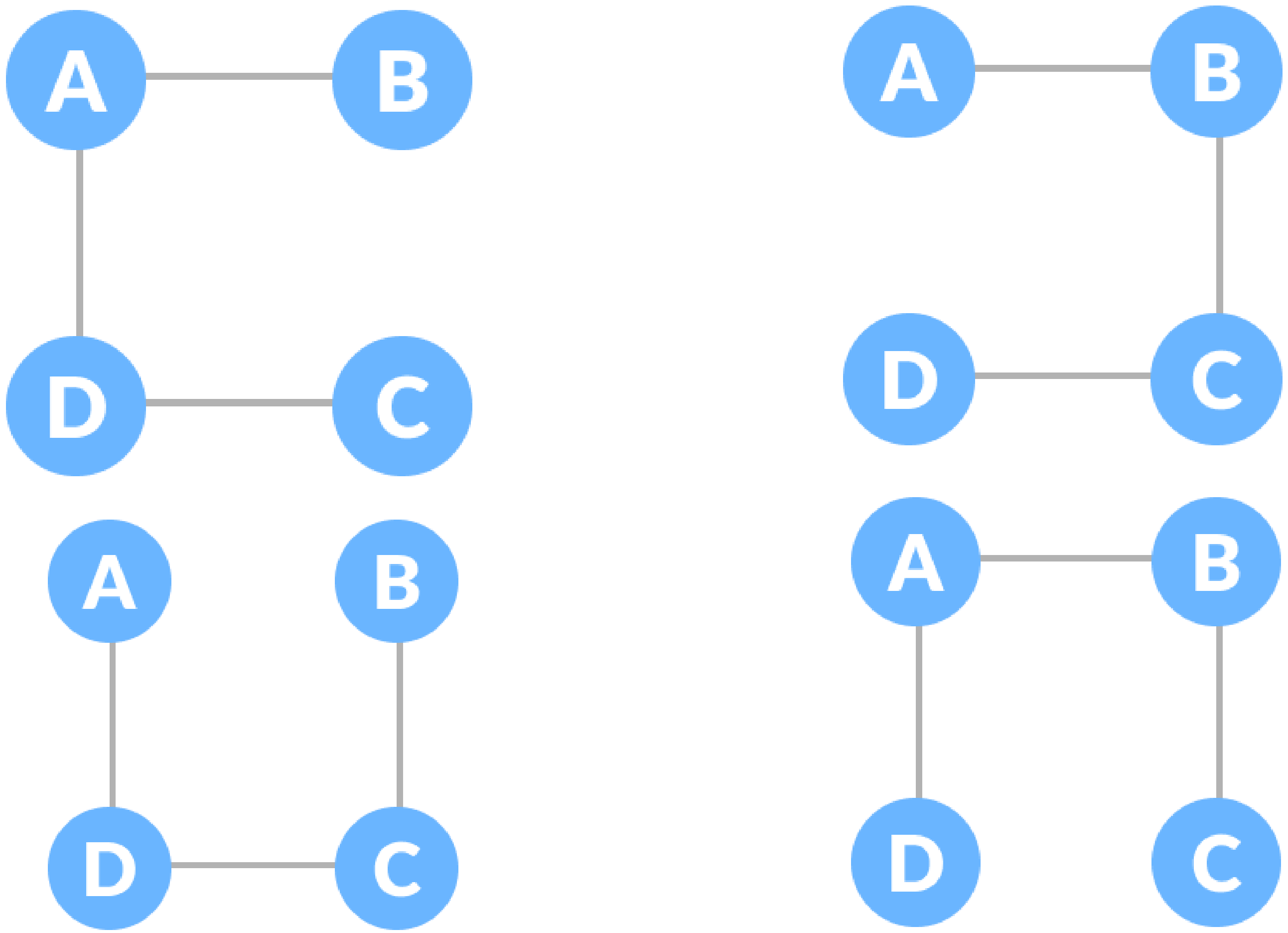
The edges may or may not have weights assigned to them.

The total number of spanning trees with n vertices that can be created from a complete graph is equal to $n^{(n-2)}$.
If we have $n = 4$, the maximum number of possible spanning trees is equal to $4^{4-2} = 16$.
Thus, 16 spanning trees can be formed from a complete graph with 4 vertices.

Spanning Tree: Example



Normal graph

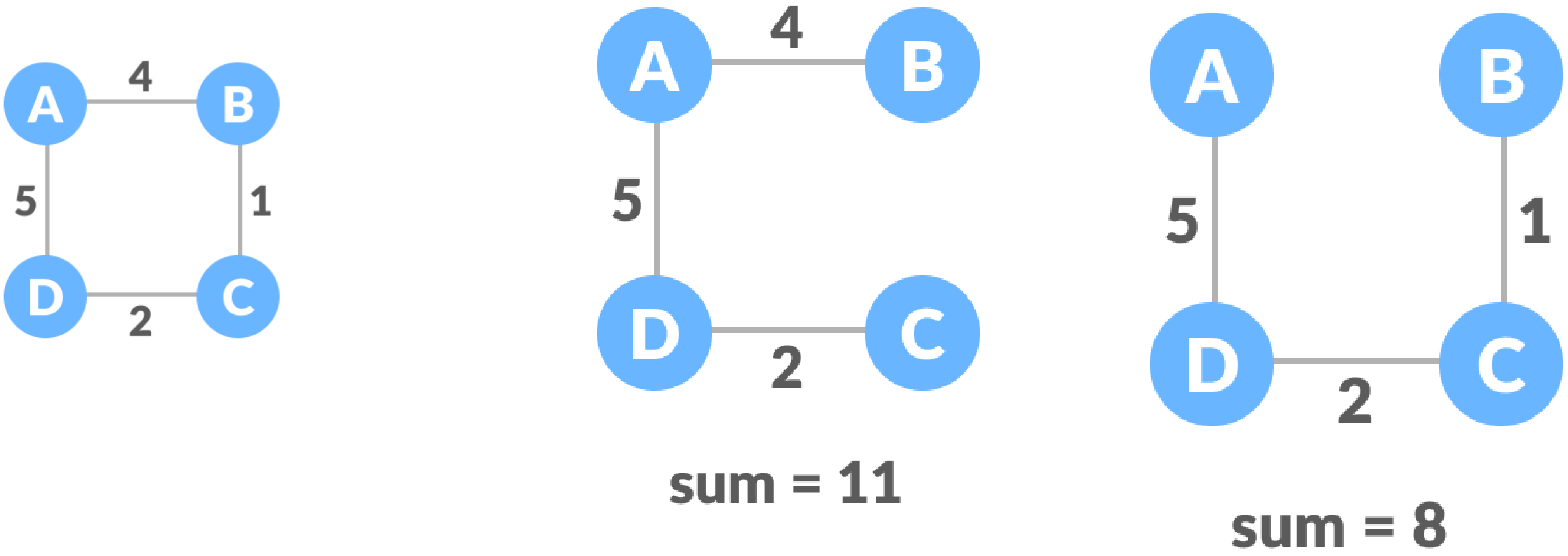


Spanning Trees

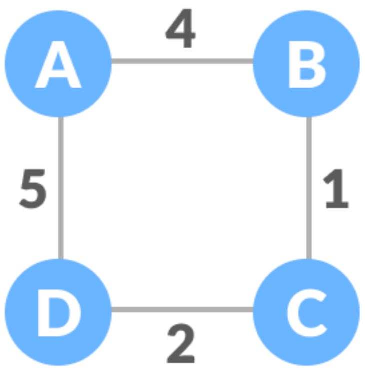
Minimum Spanning Tree

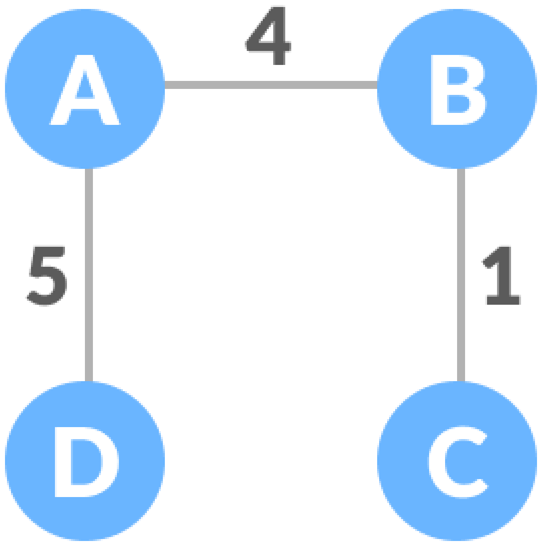
A minimum spanning tree is a spanning tree in which the sum of the weight of the edges is as minimum as possible.

Example of a Spanning Tree

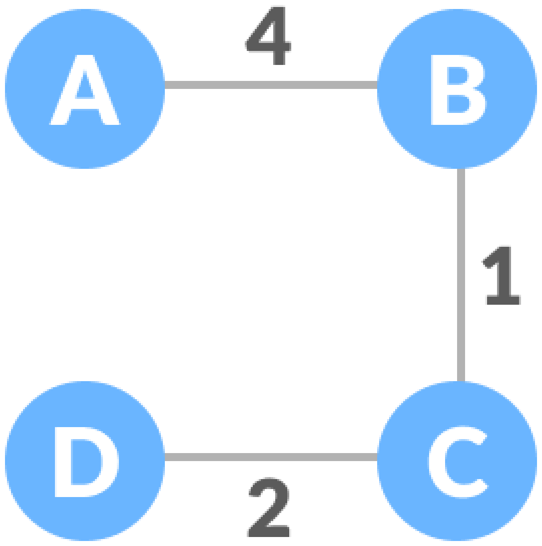


Example of a Spanning Tree

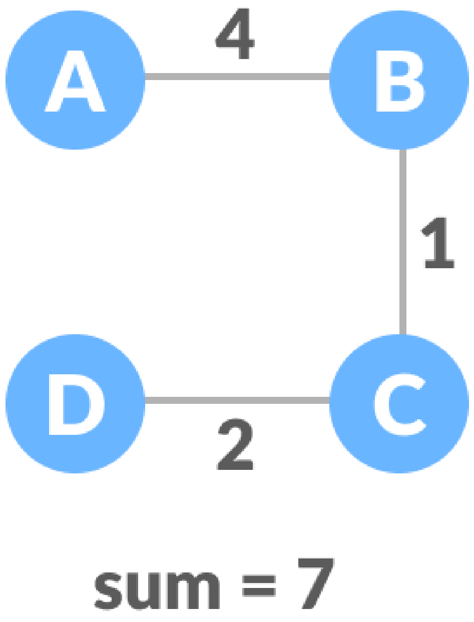




sum = 10



sum = 7



Minimum spanning tree

Spanning Tree Applications

- Computer Network Routing Protocol
- Cluster Analysis
- Civil Network Planning

The minimum spanning tree from a graph is found using the following algorithms:

- Prim's Algorithm
- Kruskal's Algorithm

Prim's Algorithm

Prim's algorithm is a minimum spanning tree algorithm that takes a graph as input and finds the subset of the edges of that graph which

- form a tree that includes every vertex
- has the minimum sum of weights among all the trees that can be formed from the graph

How Prim's algorithm works

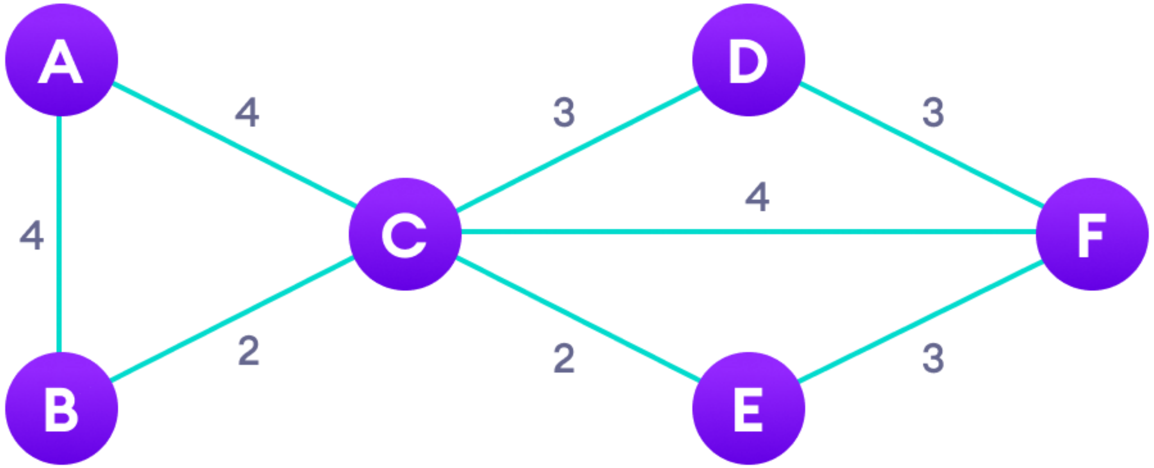
It falls under a class of algorithms called greedy algorithms that find the local optimum in the hopes of finding a global optimum.

We start from one vertex and keep adding edges with the lowest weight until we reach our goal.

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

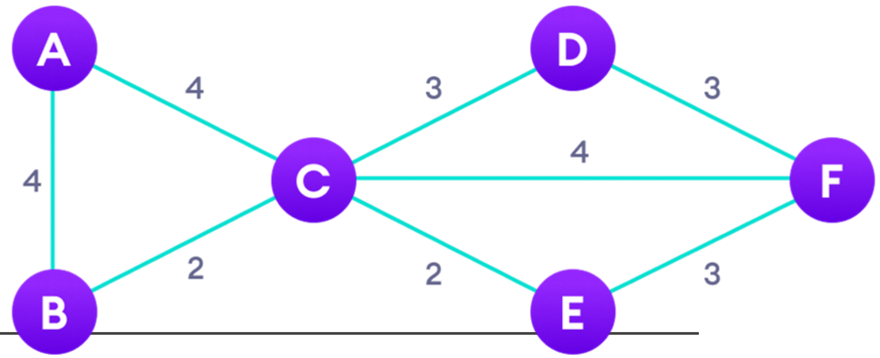
- Initialize the minimum spanning tree with a vertex chosen at random.
- Find all the edges that connect the tree to new vertices, find the minimum and add it to the tree
- Keep repeating step 2 until we get a minimum spanning tree

Example of Prim's algorithm



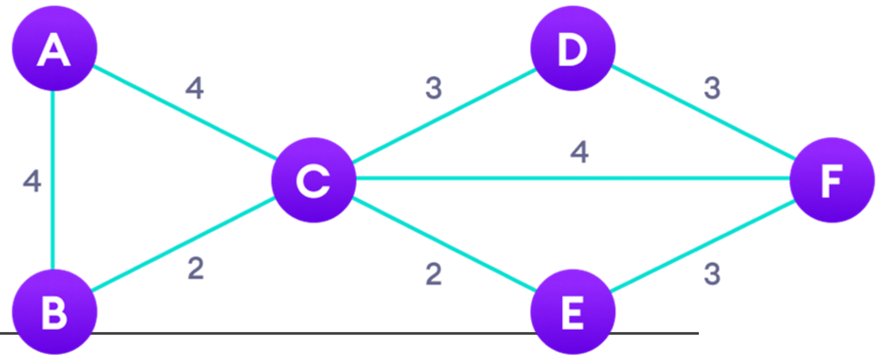
Step: 1

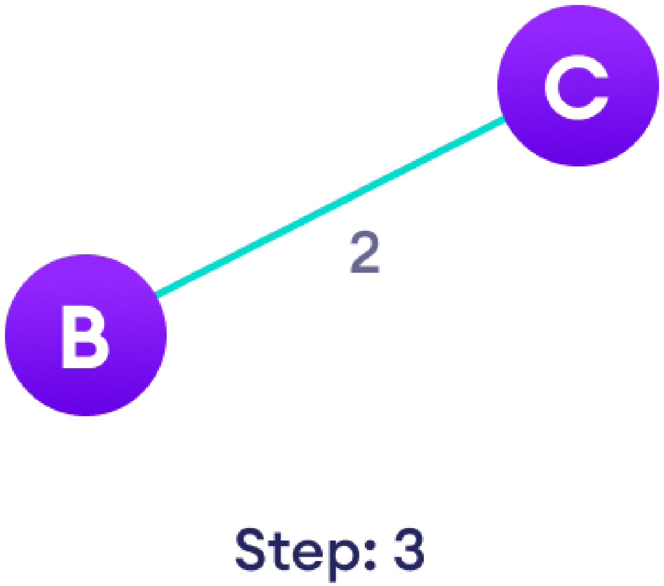
Example of Prim's algorithm



Choose a vertex

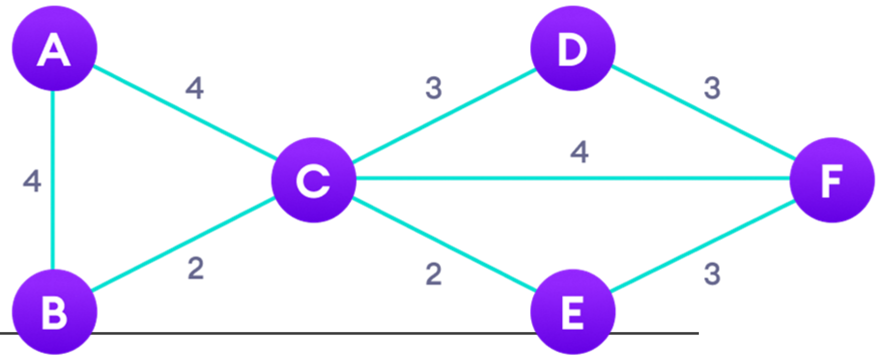
Example of Prim's algorithm

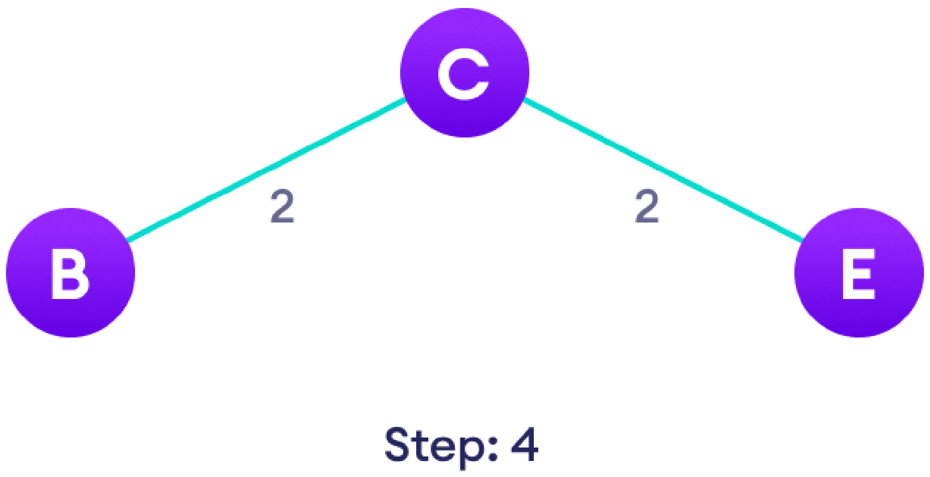




Choose the shortest edge from this vertex and add it

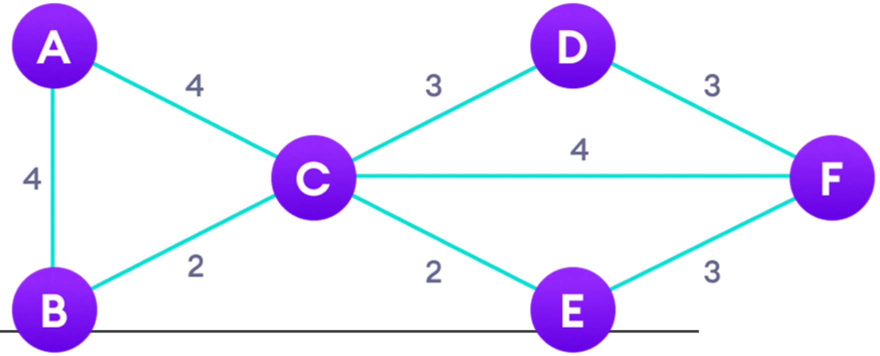
Example of Prim's algorithm



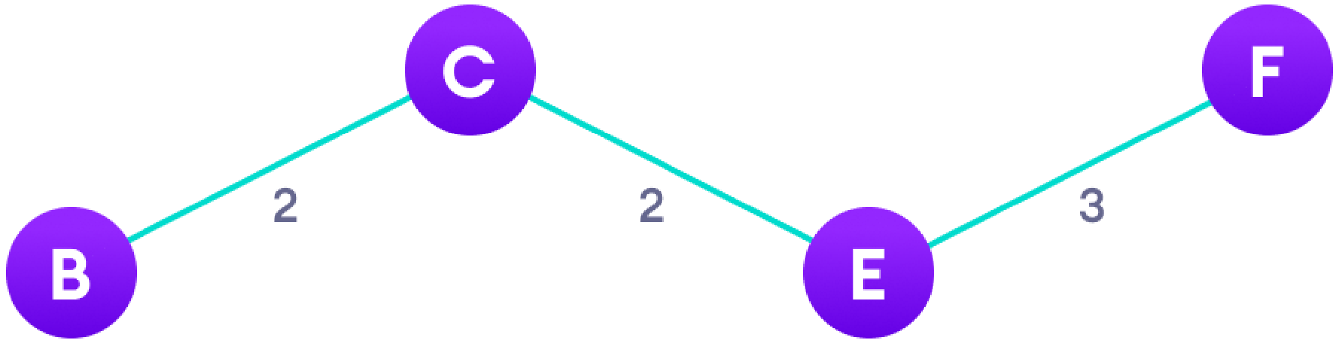


Choose the nearest vertex not yet in the solution

Example of Prim's algorithm

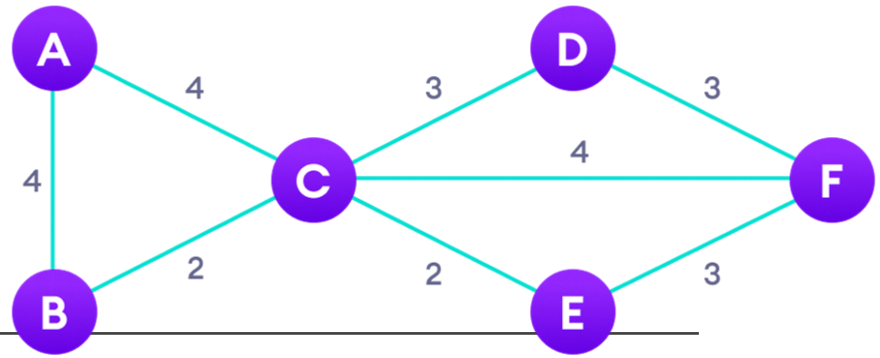


Choose the nearest vertex not yet in the solution, if there are multiple choices, choose one at random

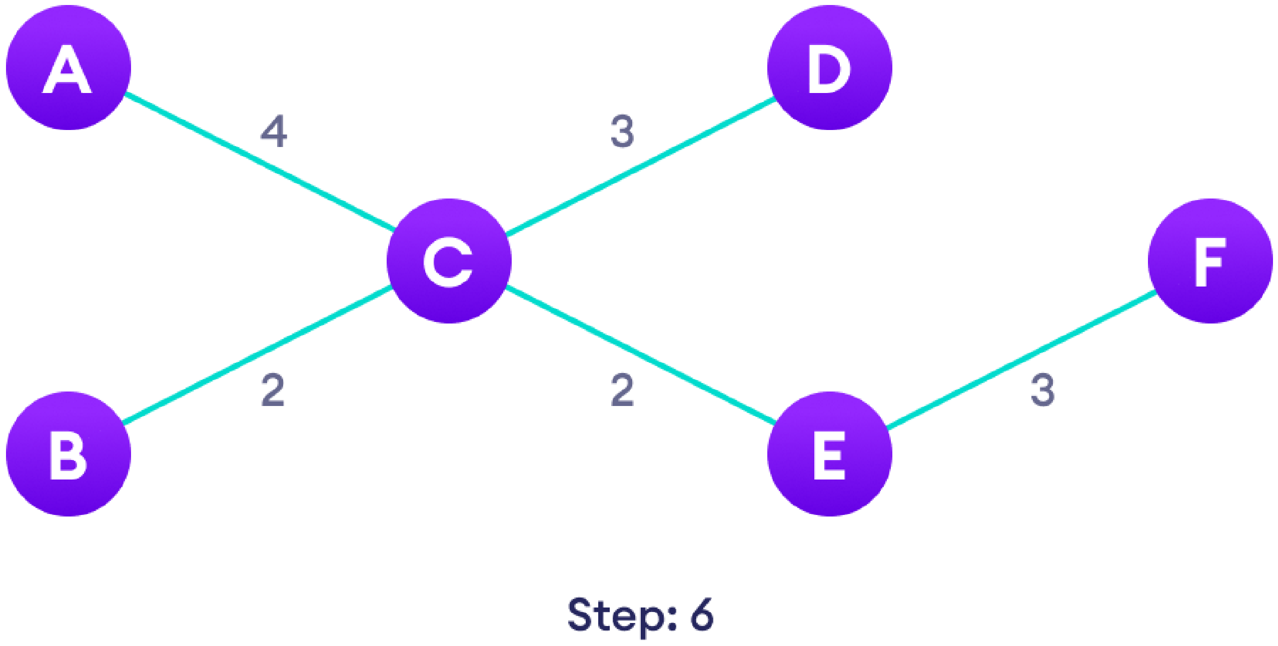


Step: 5

Example of Prim's algorithm



Repeat until you have a spanning tree



Prim's Algorithm pseudocode

The pseudocode for prim's algorithm shows how we create two sets of vertices U and V-U. U contains the list of vertices that have been visited and V-U the list of vertices that haven't. One by one, we move vertices from set V-U to set U by connecting the least weight edge.

```
T = ∅;
U = { 1 };
while (U ≠ V)
    let (u, v) be the lowest cost edge such that u ∈ U and v ∈ V - U;
    T = T ∪ {(u, v)}
    U = U ∪ {v}
```

Credits and Acknowledgements

https://www.gatevidyalay.com

https://www.programiz.com/