Lecture#11

Data Structures

Dr. Abu Nowshed Chy

Department of Computer Science and Engineering

University of Chittagong

February 24, 2025

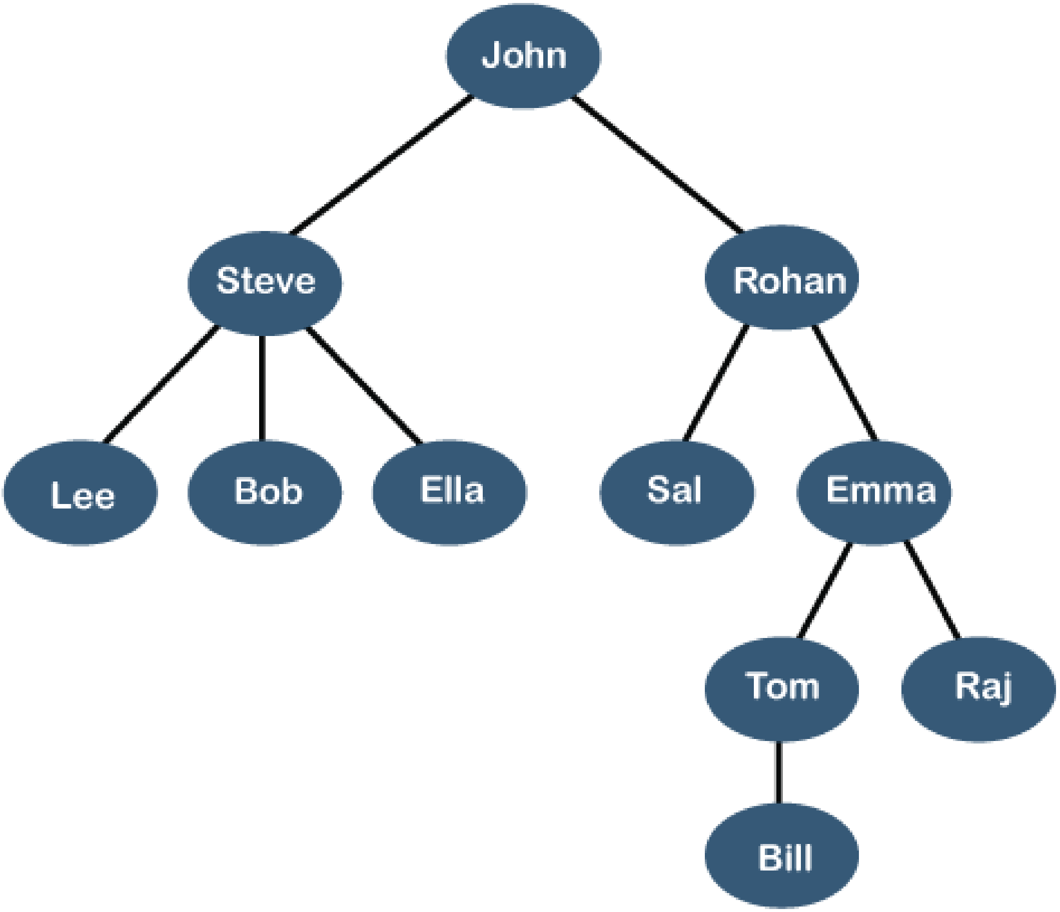
Faculty Profile

Tree

Tree

- A tree is an undirected graph which is connected and acyclic. It is easy to show that if graph *G*
  - G (V,E) is connected.
  - G (V,E) is acyclic.
  - $|E| = |V| - 1$

Tree



Tree

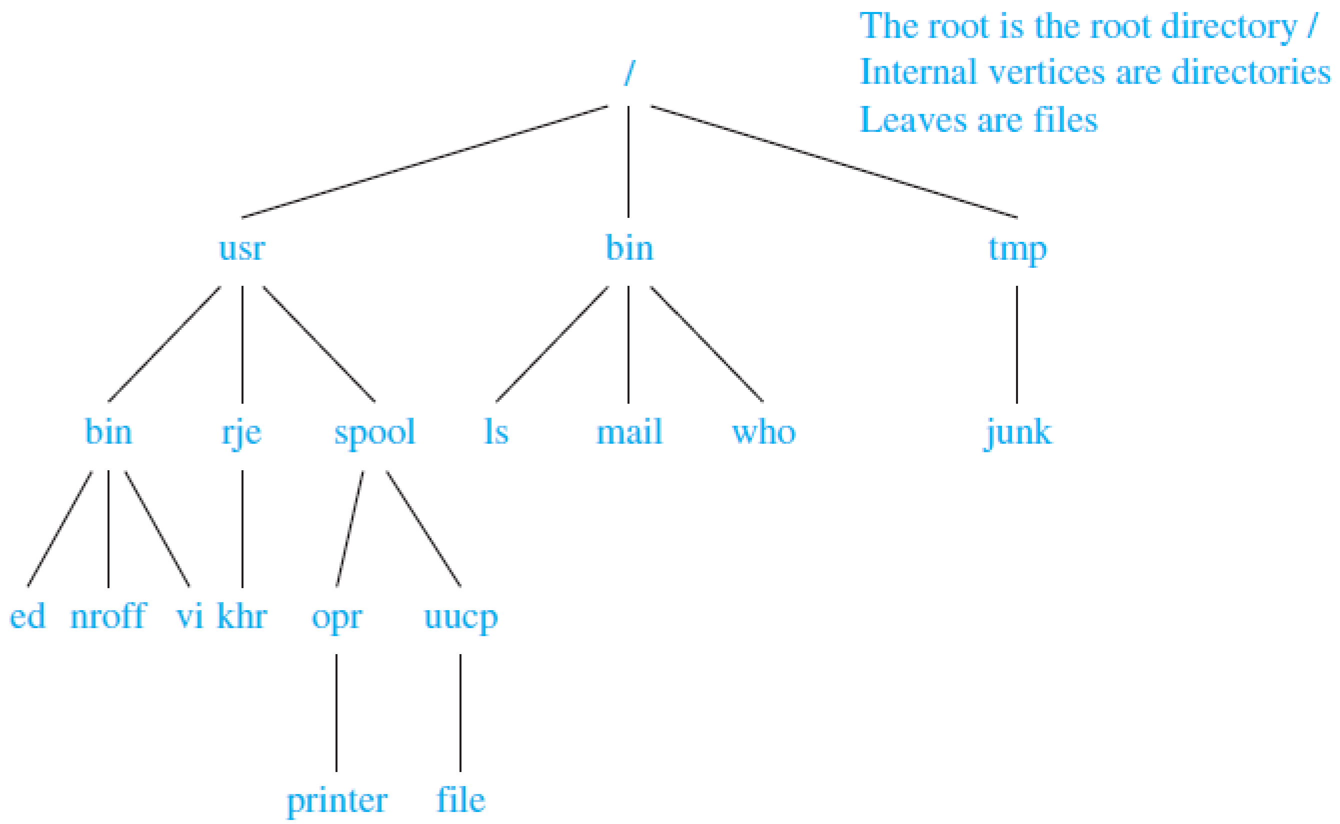


FIGURE 11 **A Computer File System.**

Tree Traversal

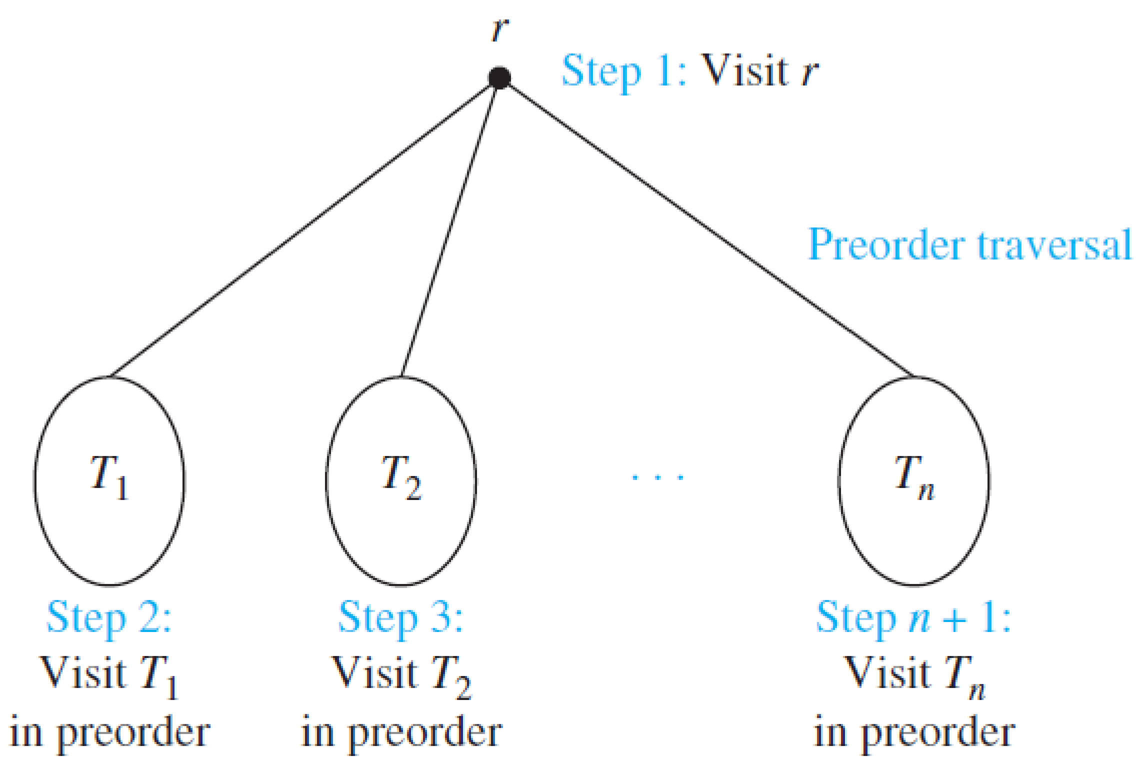


FIGURE 2 Preorder Traversal.

Tree Traversal

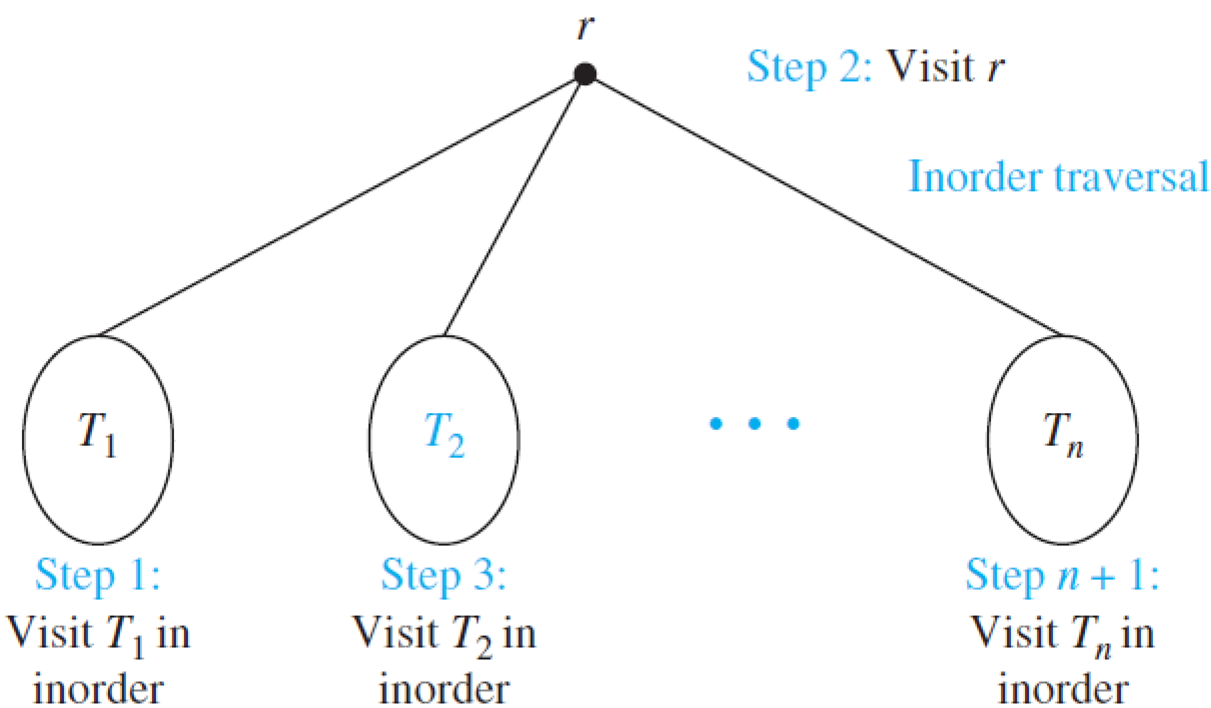


FIGURE 5 **Inorder Traversal.**

Tree Traversal

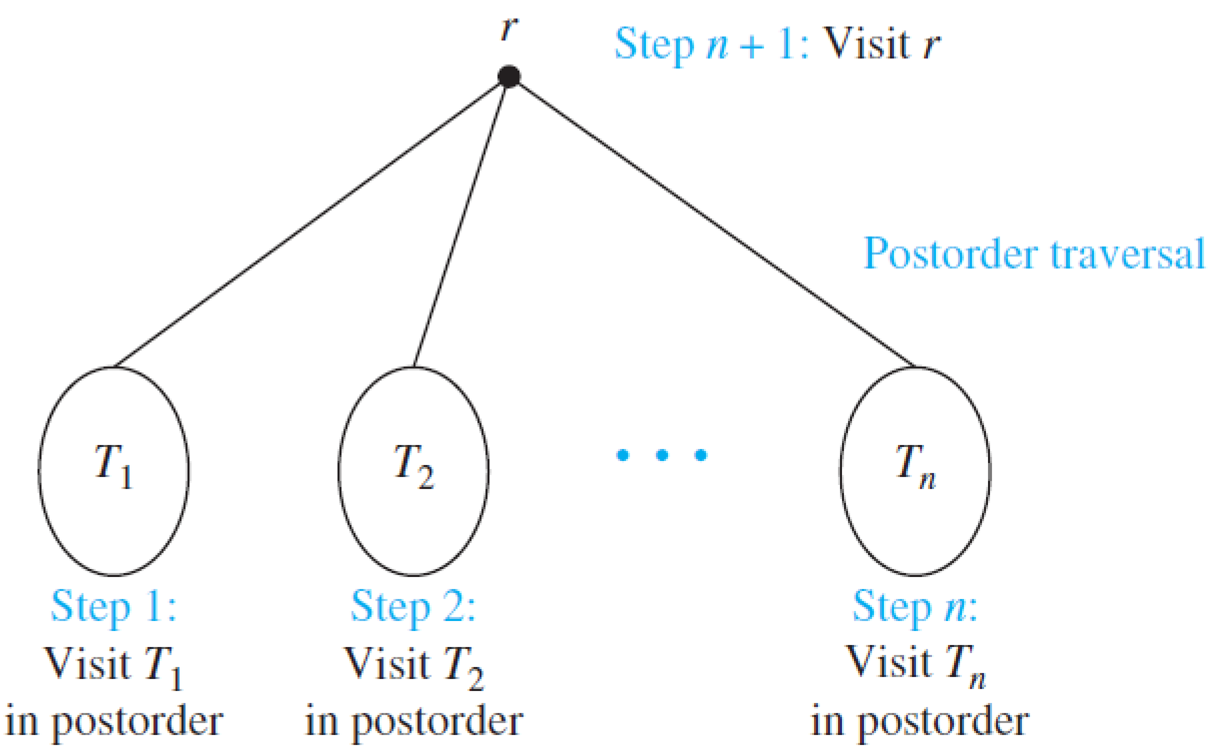
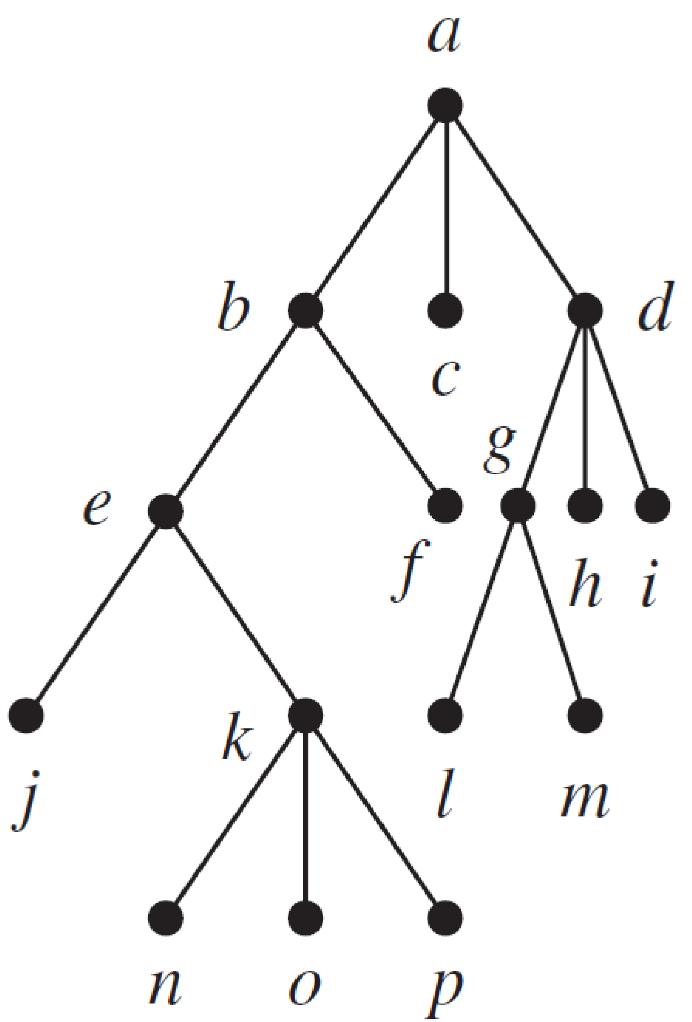


FIGURE 7 Postorder Traversal.

Tree Traversal



Tree Traversal

$a \quad b \quad e \quad j \quad k \quad n \quad o \quad p \quad f \quad c \quad d \quad g \quad l \quad m \quad h \quad i$

FIGURE 4 **The Preorder Traversal of *T*.**

Tree Traversal

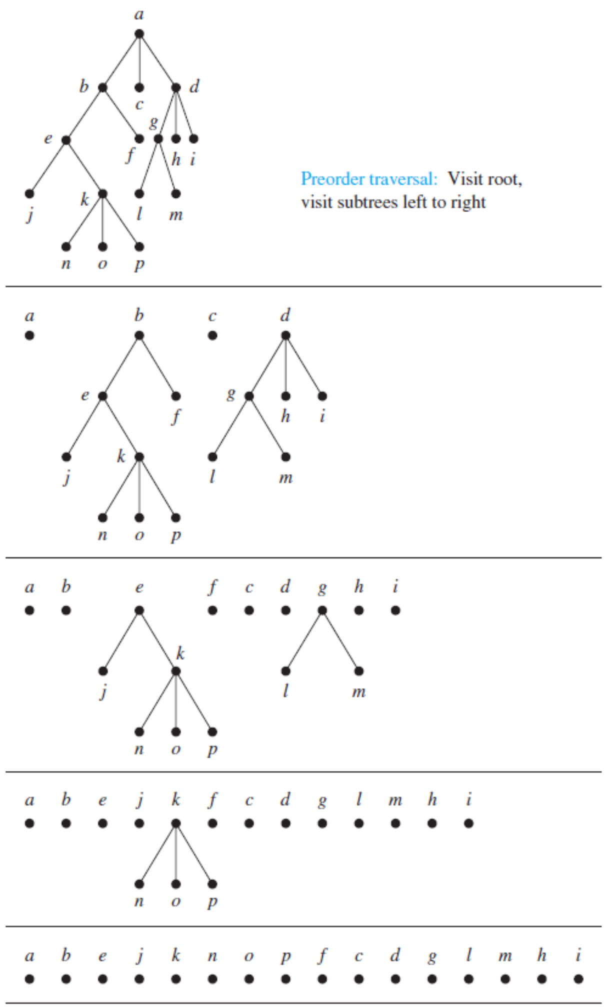


FIGURE 4 The Preorder Traversal of T.

Tree Traversal

j e n k o p b f a c l g m d h i

FIGURE 6 The Inorder Traversal of *T*.

Tree Traversal

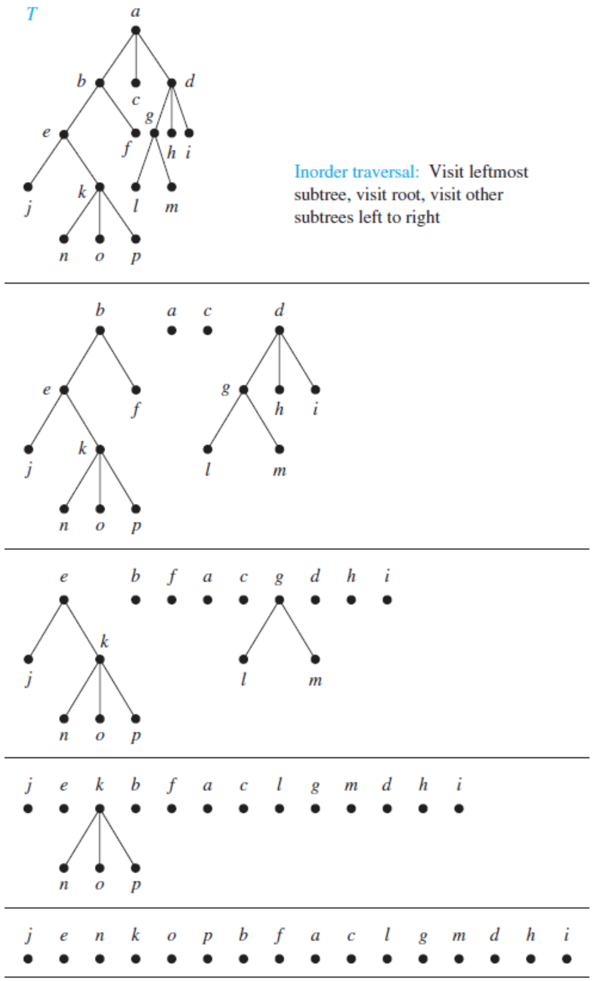


FIGURE 6 The Inorder Traversal of *T*.

Tree Traversal

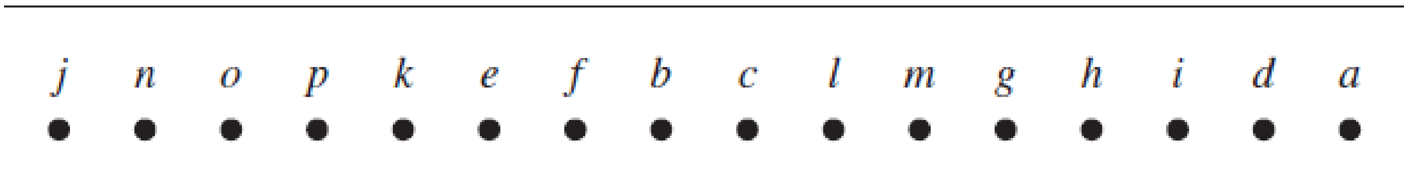


FIGURE 8 **The Postorder Traversal of *T*.**

Tree Traversal

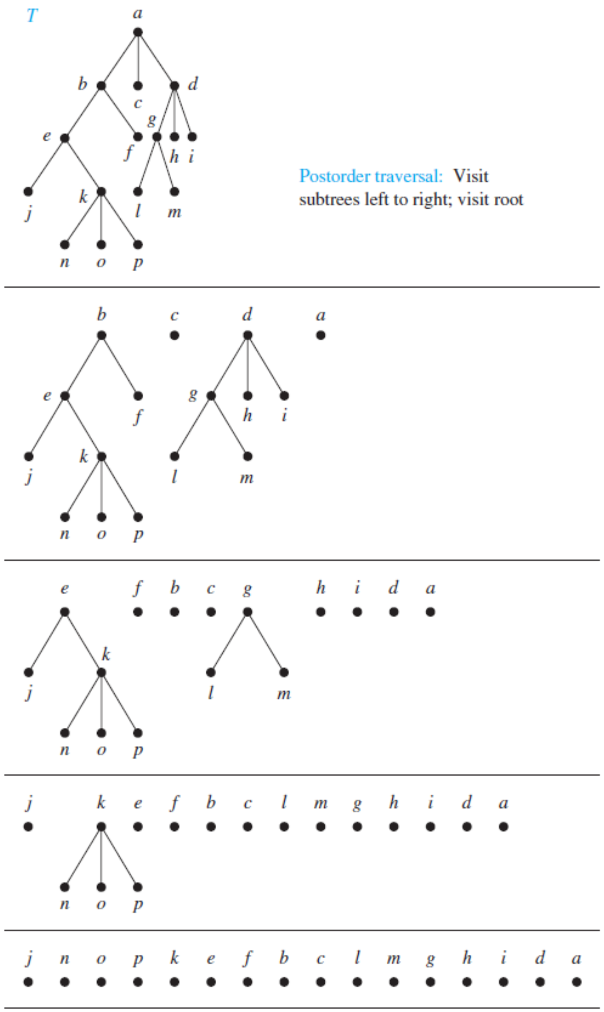
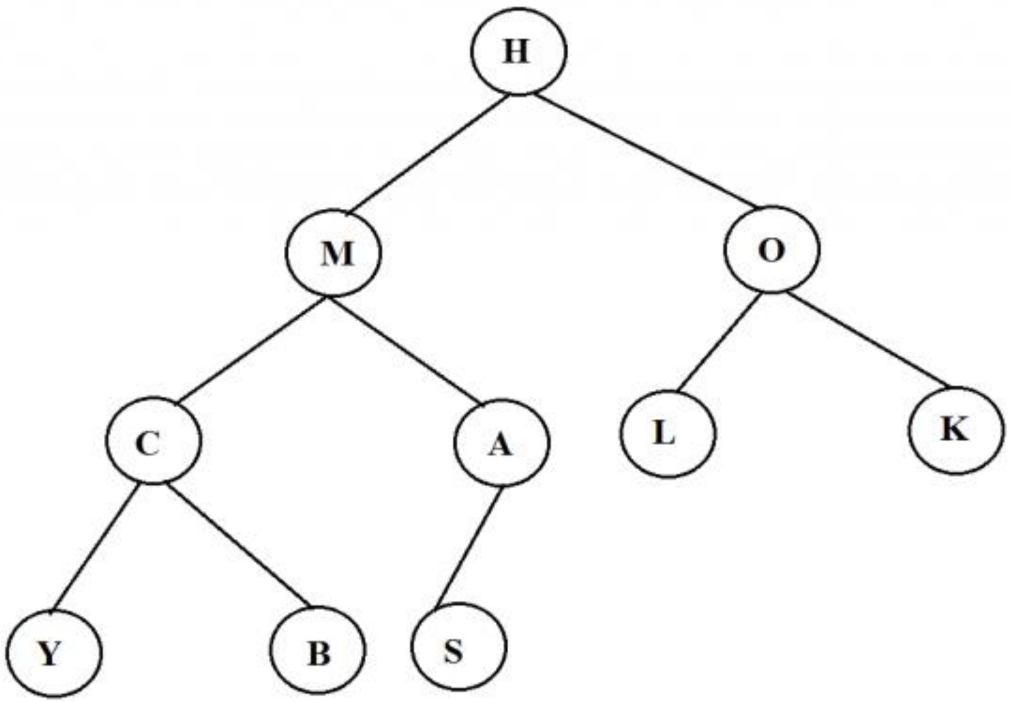


FIGURE 8 The Postorder Traversal of T.

Tree Traversal

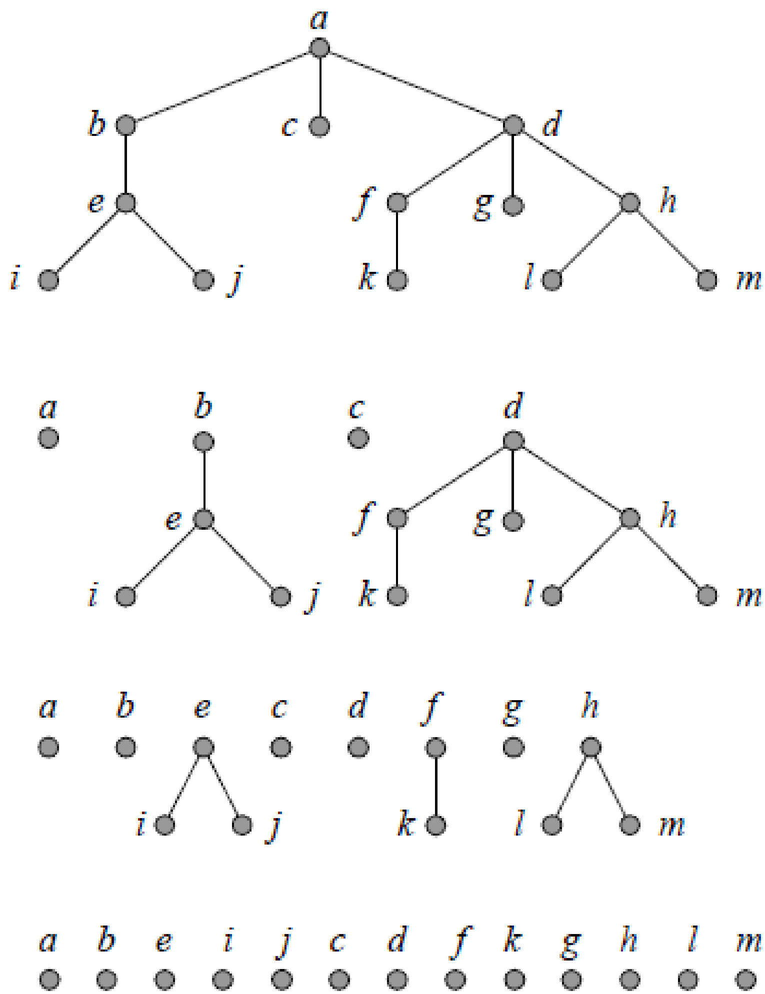


InOrder traversal: Y C B M S A H L O K

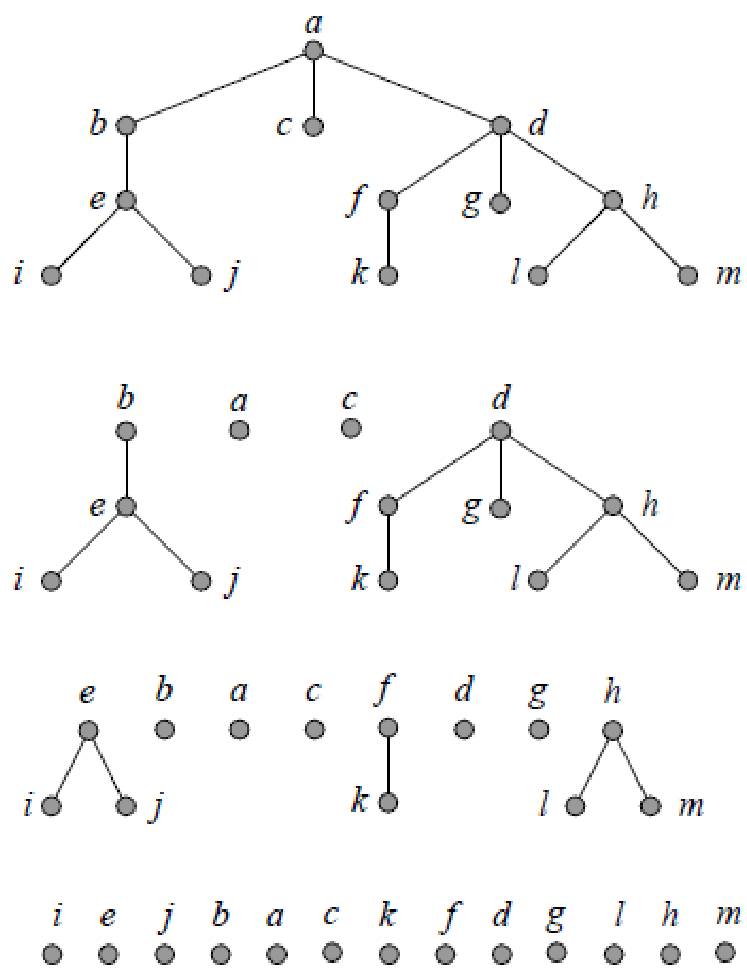
Preorder traversal: H M C Y B A S O L K

Postorder traversal: Y B C S A M L K O H

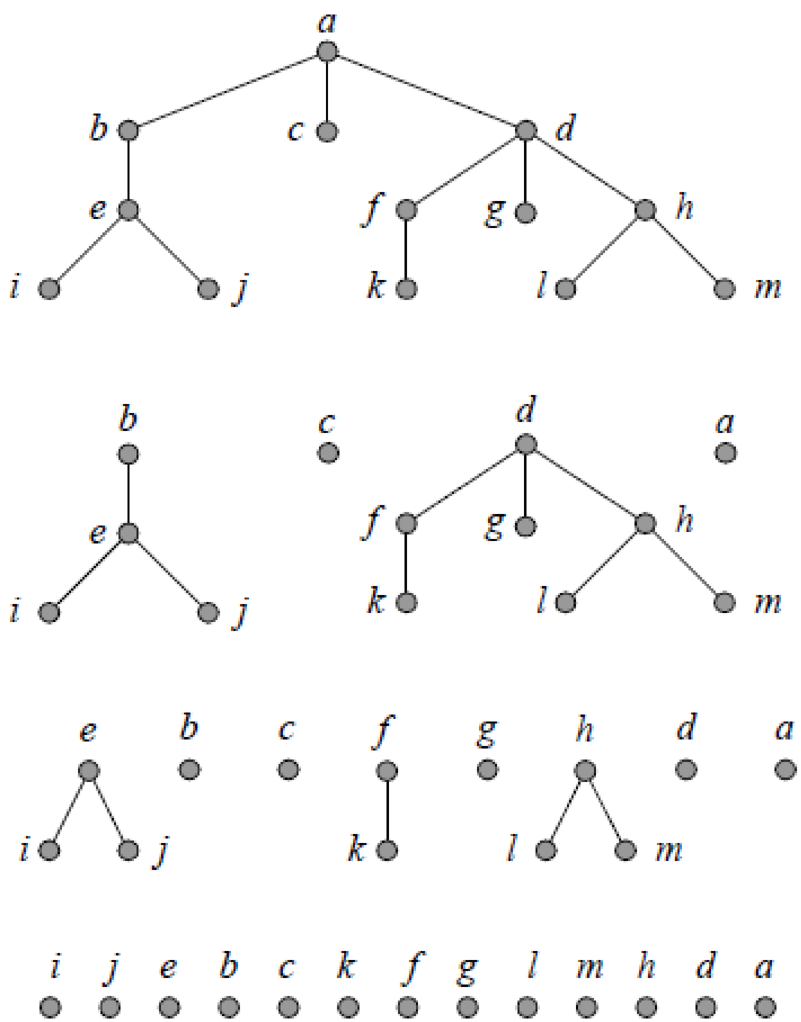
Tree Traversal (Preorder)



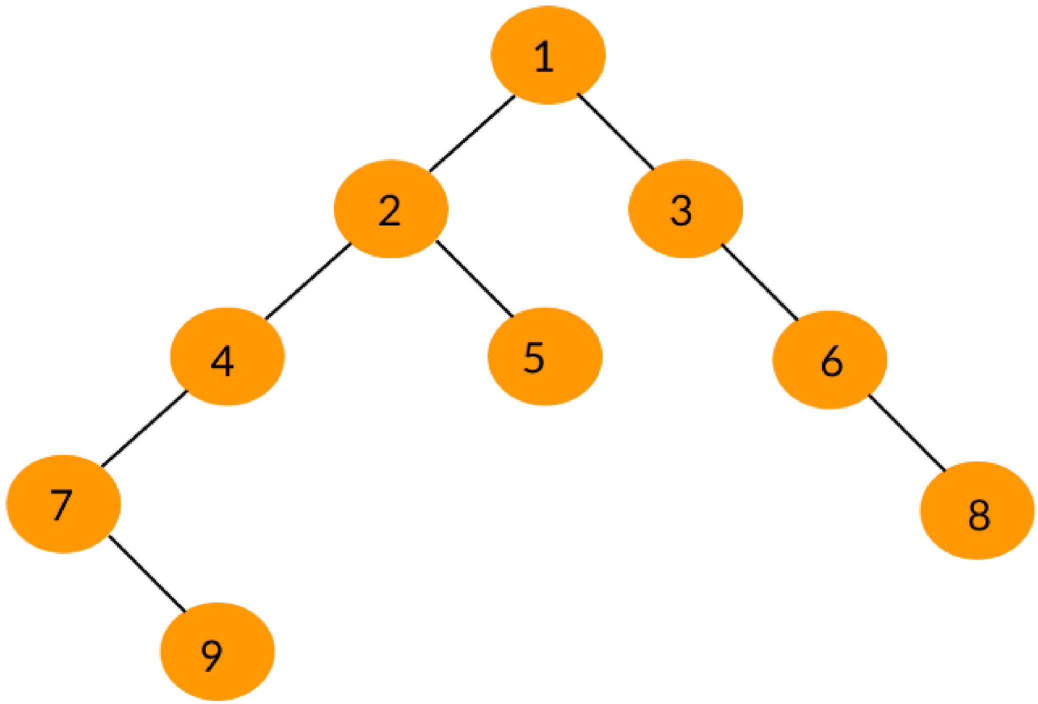
Tree Traversal (Inorder)



Tree Traversal (Postorder)



Tree Traversal

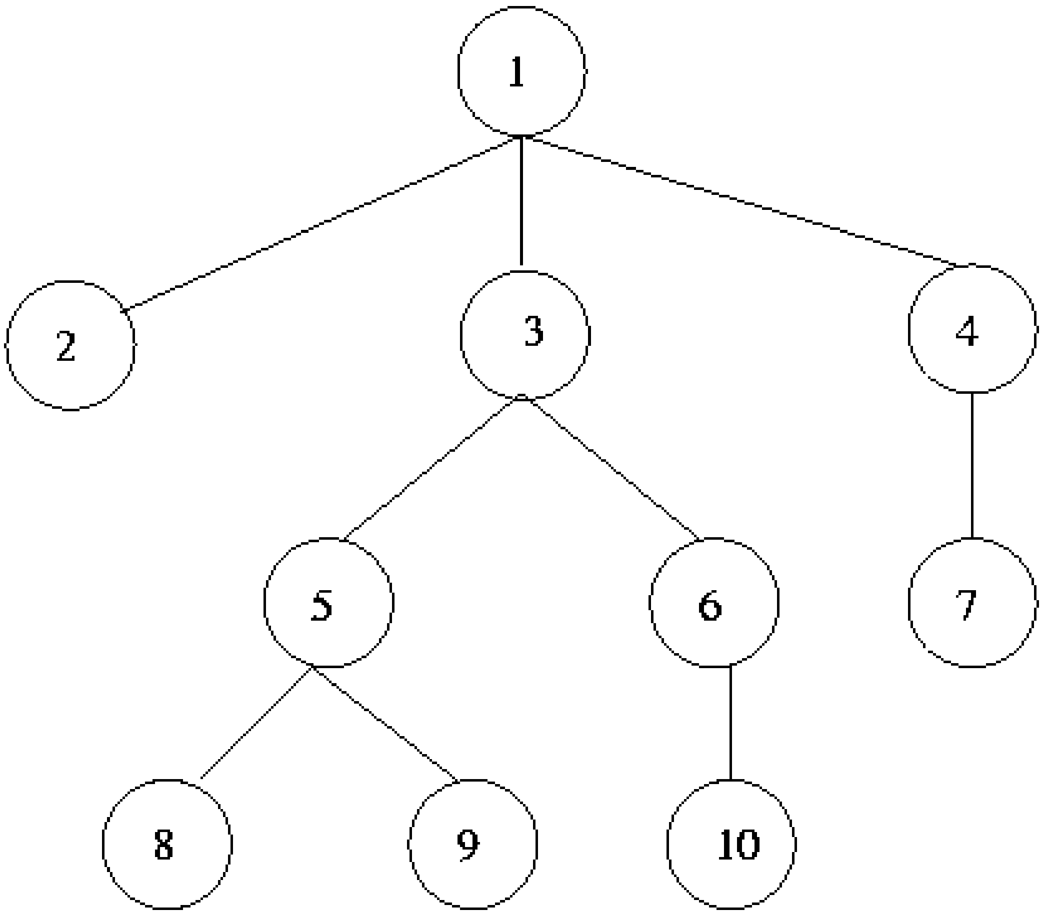


Inorder Traversal: 7 9 4 2 5 1 3 6 8

Preorder Traversal: 1 2 4 7 9 5 3 6 8

Postorder Traversal: 9 7 4 5 2 8 6 3 1

Tree Traversal

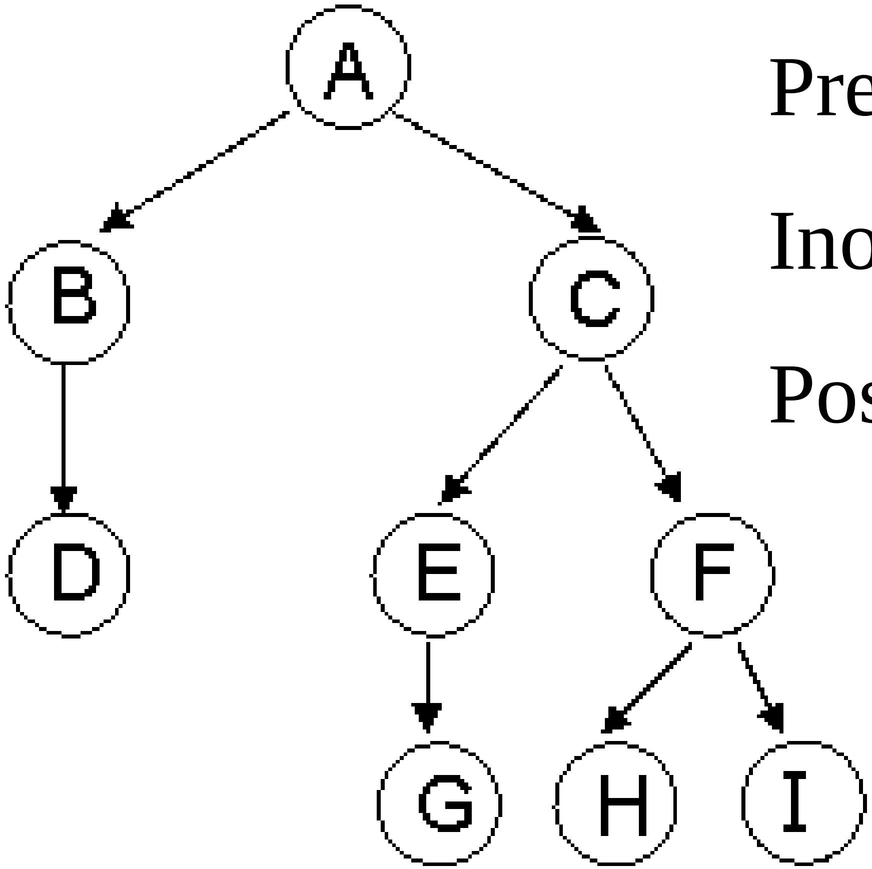


Preorder 1,2,3,5,8,9,6,10,4,7

Postorder 2,8,9,5,10,6,3,7,4,1

Inorder 2,1,8,5,9,3,10,6,7,4

Tree Traversal

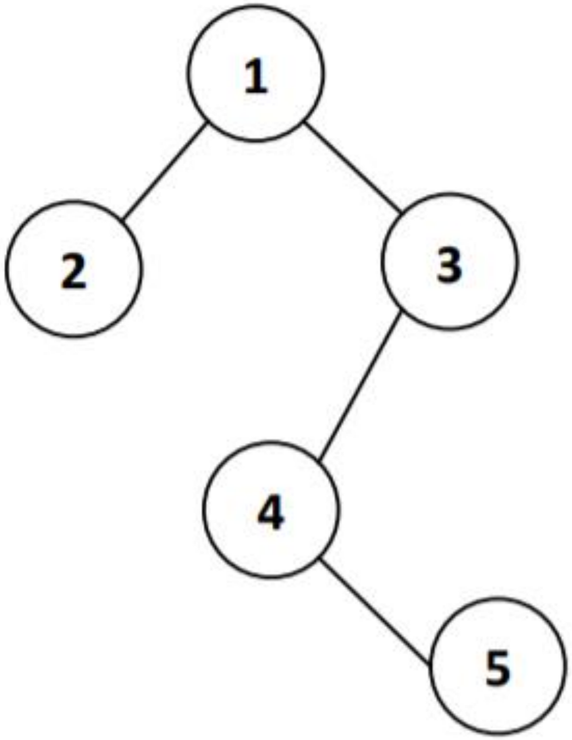


Preorder: A B D C E G F H I

Inorder: D B A G E C H F I

Postorder: D B G E H I F C A

Tree Traversal



Inorder: 2 1 4 5 3

Preorder: 1 2 3 4 5

Post order: 2 5 4 3 1

Tree Traversal

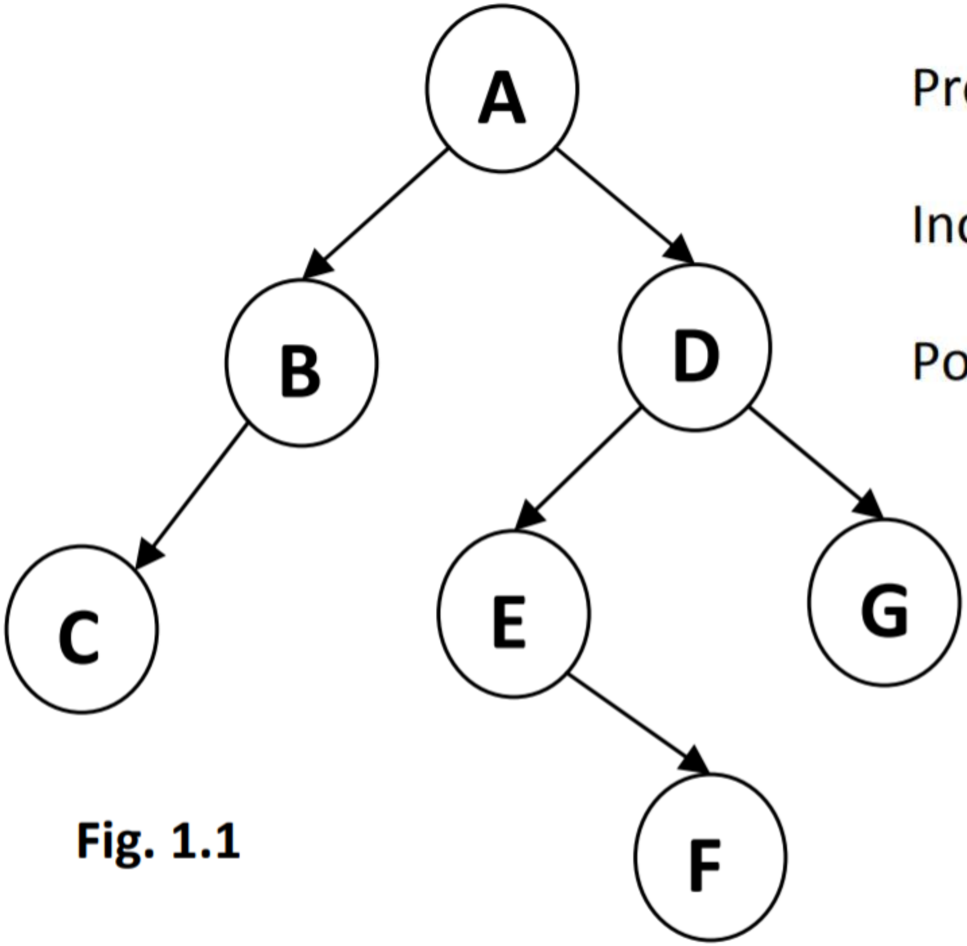


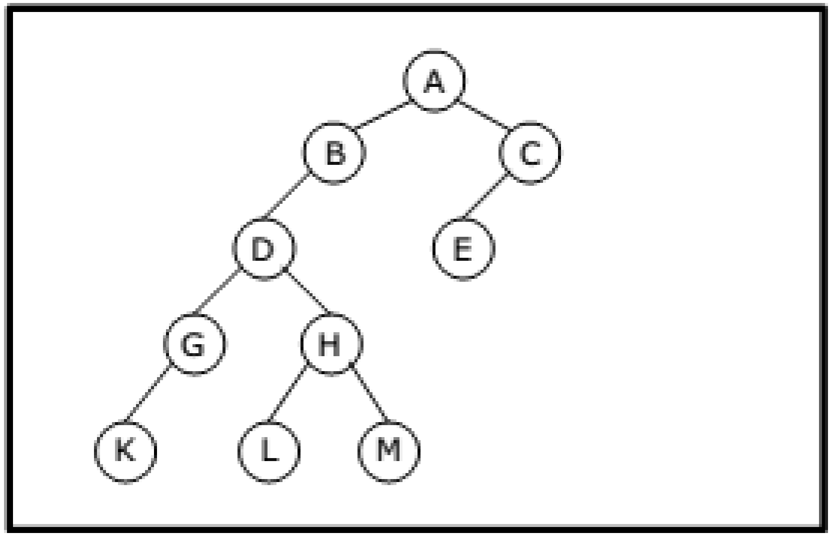
Fig. 1.1

Preorder traversal : A B C D E F G

Inorder traversal : C B A E F D G

Postorder traversal : C B F E G D A

Tree Traversal

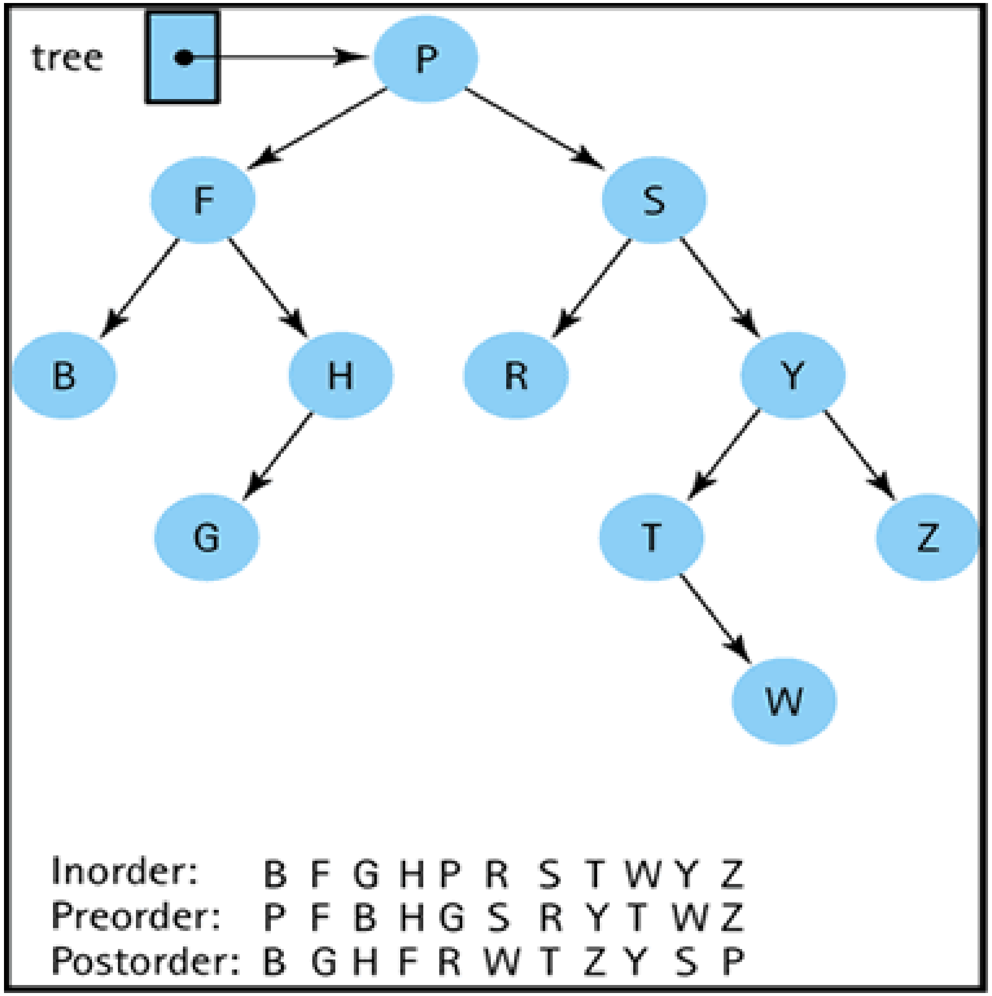


Binary Tree

- Preorder traversal yields:
  A, B, D, G, K, H, L, M, C, E
- Postorder travarsal yields:
  K, G, L, M, H, D, B, E, C, A
- Inorder travarsal yields:
  K, G, D, L, H, M, B, A, E, C

Pre, Post and Inorder Traversing

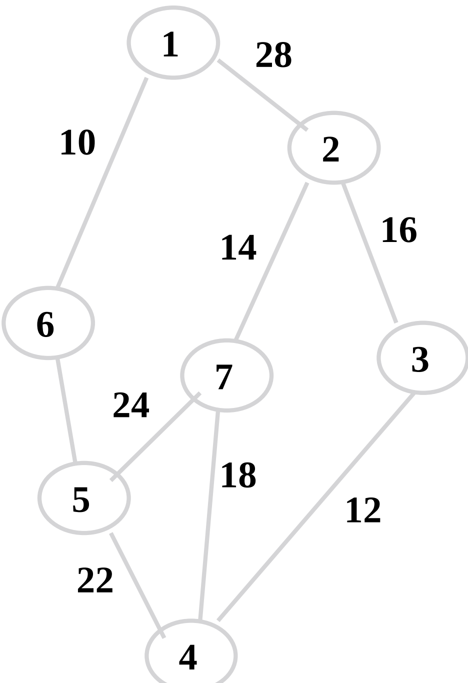
Tree Traversal



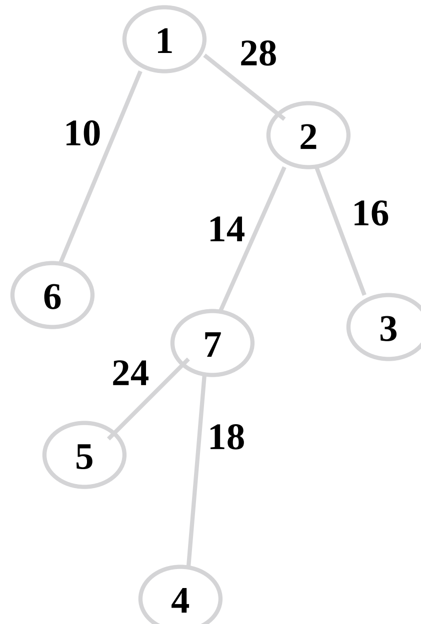
Minimum Spanning Tree (MST)

- A *spanning tree* is a tree that contains all of the vertices in the graph and enough of its edges to form a tree.
- The *minimum spanning tree* of a network is a spanning tree in which the sum of its edge weights is guaranteed to be minimal.
- It is possible to have more than one minimum spanning tree – however the weights of the different MSTs will be the same.
- The *minimum spanning tree* in a weighted graph *G(V,E)* is one which has the smallest weight among all spanning trees in *G(V,E).*

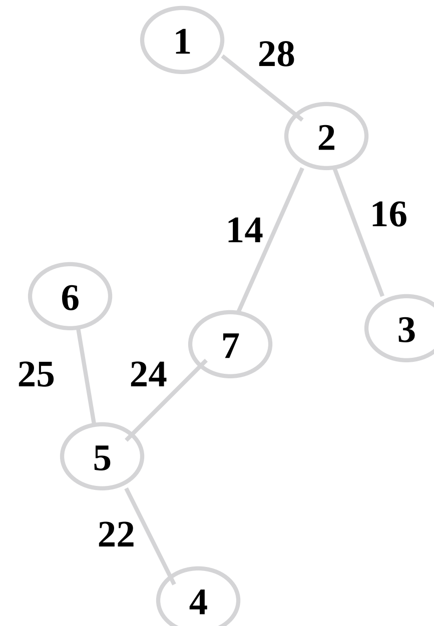
Minimum Spanning Tree (MST)



connected graph



spanning tree
cost = 110



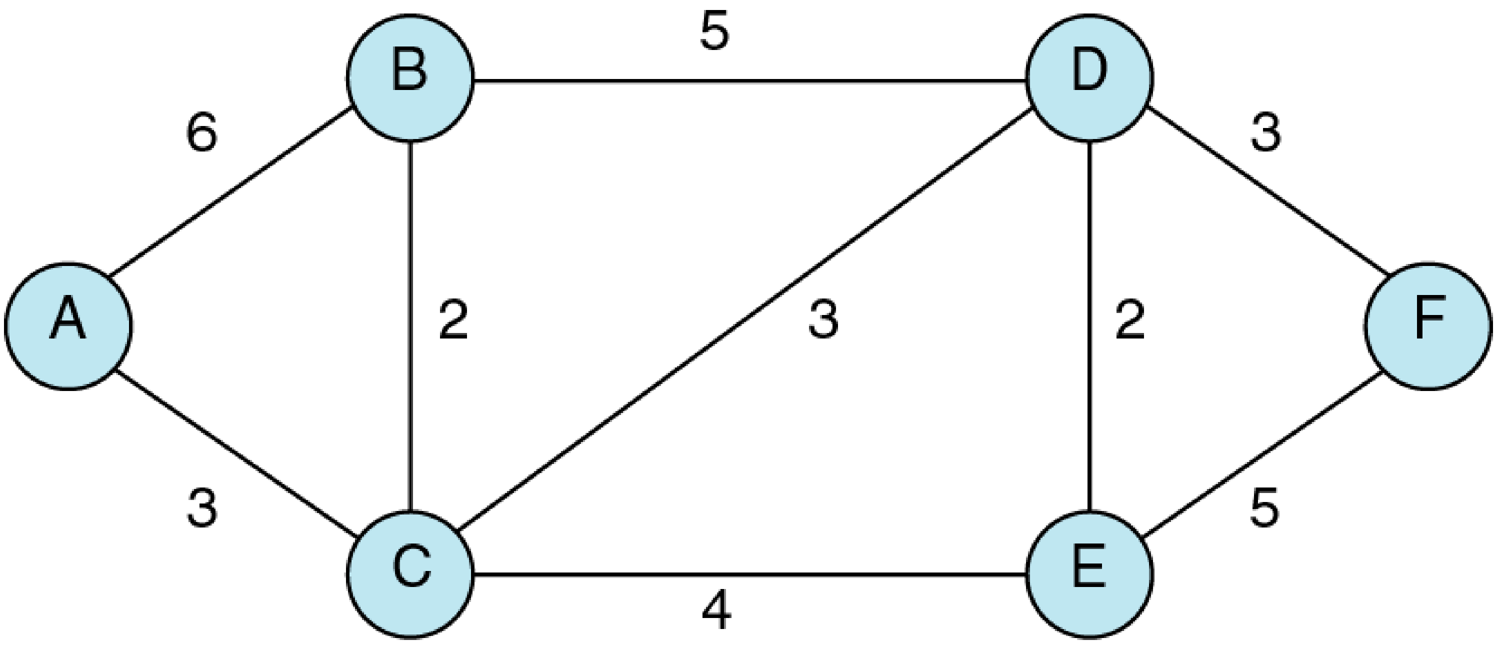
spanning tree
cost = 129

Applications

- There are many applications for minimum spanning trees, all with the requirement to minimize some aspect of the graph, such as the distance among all the vertices in the graph.
- For example, given a network of computers, we could create a graph that connects all of the computers.
- The MST gives us the shortest length of cable that can be used to connect all of the computers while ensuring that there is a path between any two computers.

Prim’s Algorithm for MST

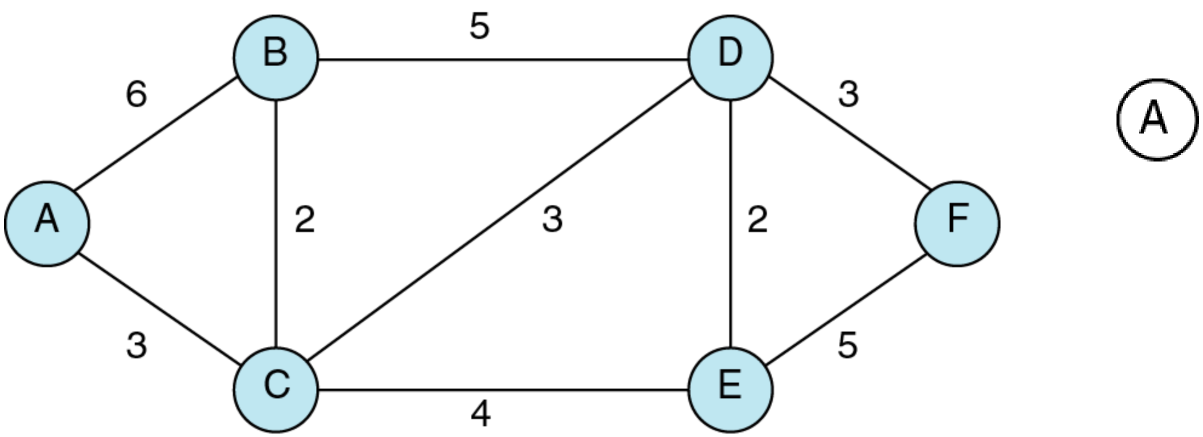
Prim's Algorithm



Prim’s Algorithm

- We can start with any vertex, so we’ll just start with A.
- Then, we add the vertex that gives the minimum-weighted edge with A.
- Our options are AB or AC – we choose AC because 3 is less than 6

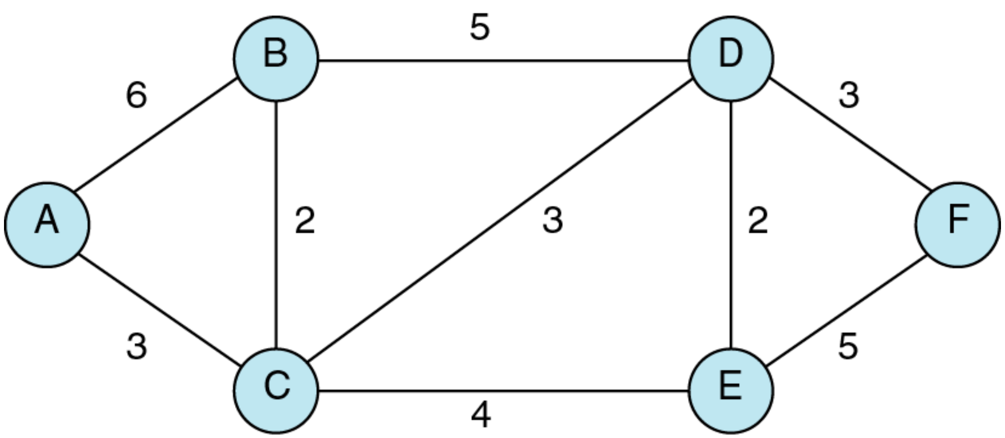
MST



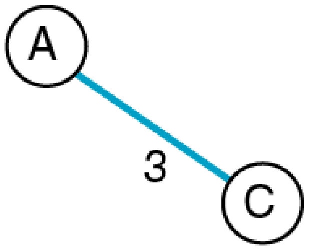
(a) Insert first vertex

Prim's Algorithm

- Now, we have vertices A and C in our MST.
- From these 2, we choose the edge with the minimum weight that connects a vertex in our MST to a vertex not already included in our MST.



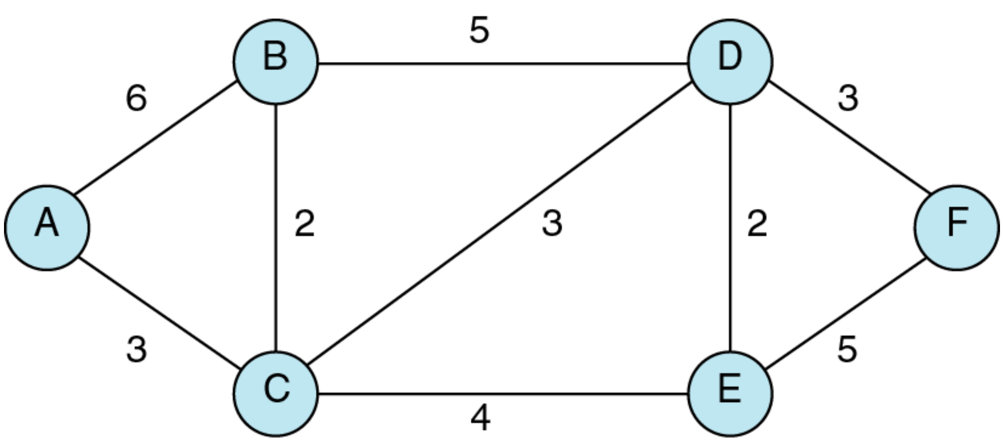
MST



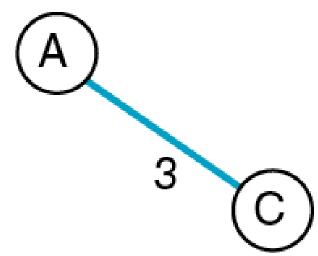
(b) Insert edge

Prim’s Algorithm

- Our options are
  - CB, weight = 2
  - CD, weight = 3
  - CE, weight = 4
  - AB, weight = 6
- We choose CB because it has the minimum weight.



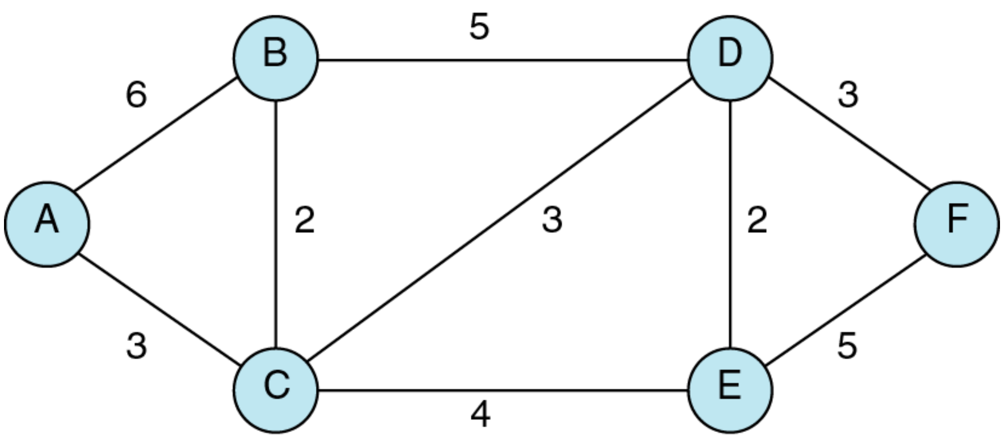
MST

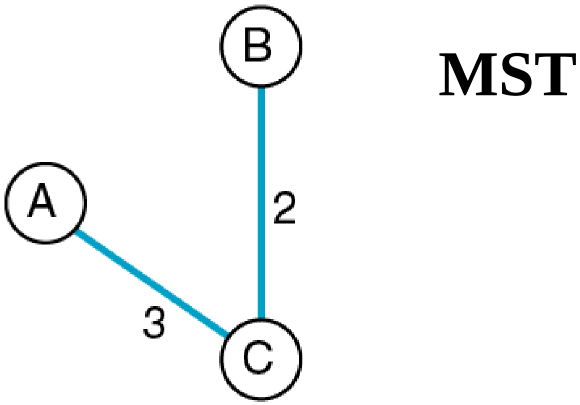


(b) Insert edge

Prim's Algorithm

- Now we continue with the same process – here are our choices:
  - BD, weight = 5
  - CD, weight = 3
  - CE, weight = 4
- We choose CD because it has the minimum weight.

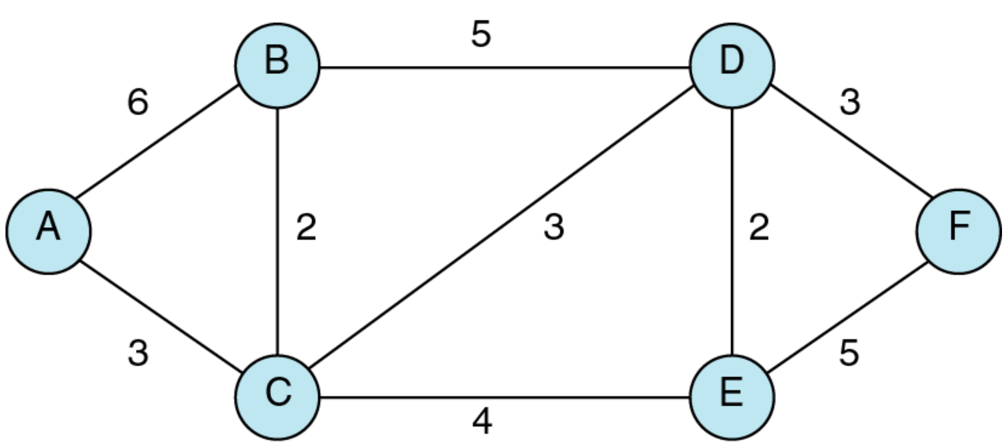


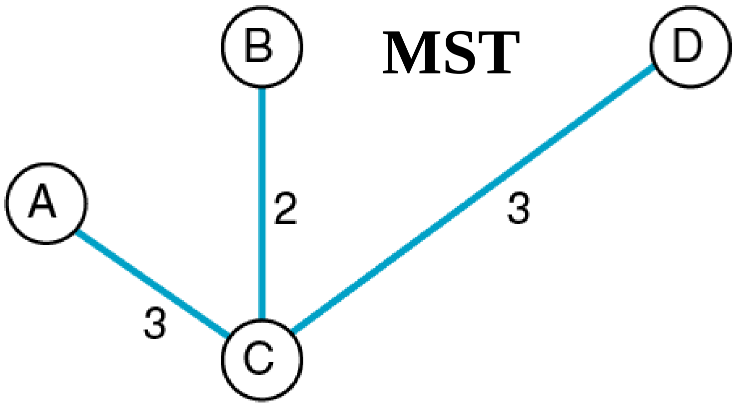


(c) Insert edge BC

Prim's Algorithm

- Now we continue with the same process – here are our choices:
  - CE, weight = 4
  - DE, weight = 2
  - DF, weight = 3
- We choose DE because it has the minimum weight.

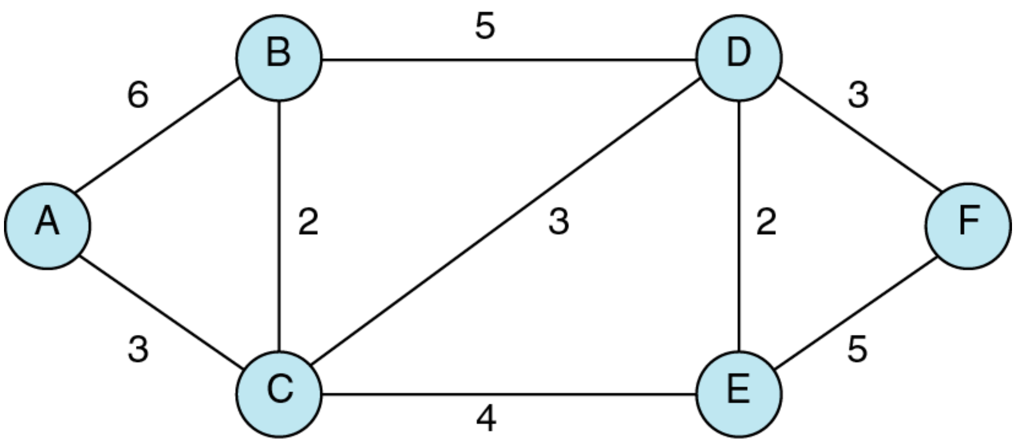


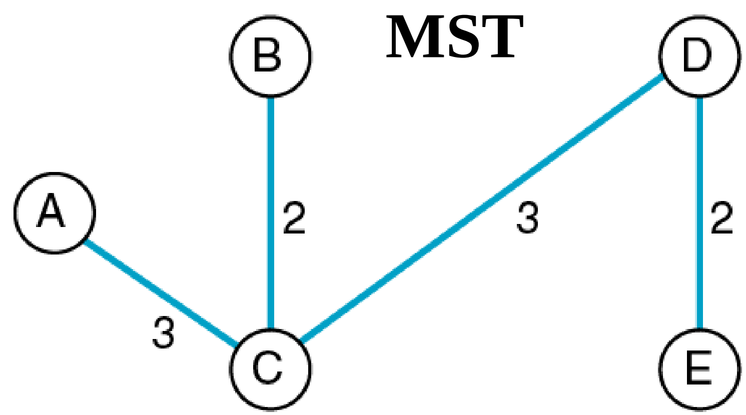


(d) Insert edge CD

Prim's Algorithm

- Now we are down to our last node. Our choices are:
  - EF, weight = 5
  - DF, weight = 3
- We choose DF because it has the minimum weight.

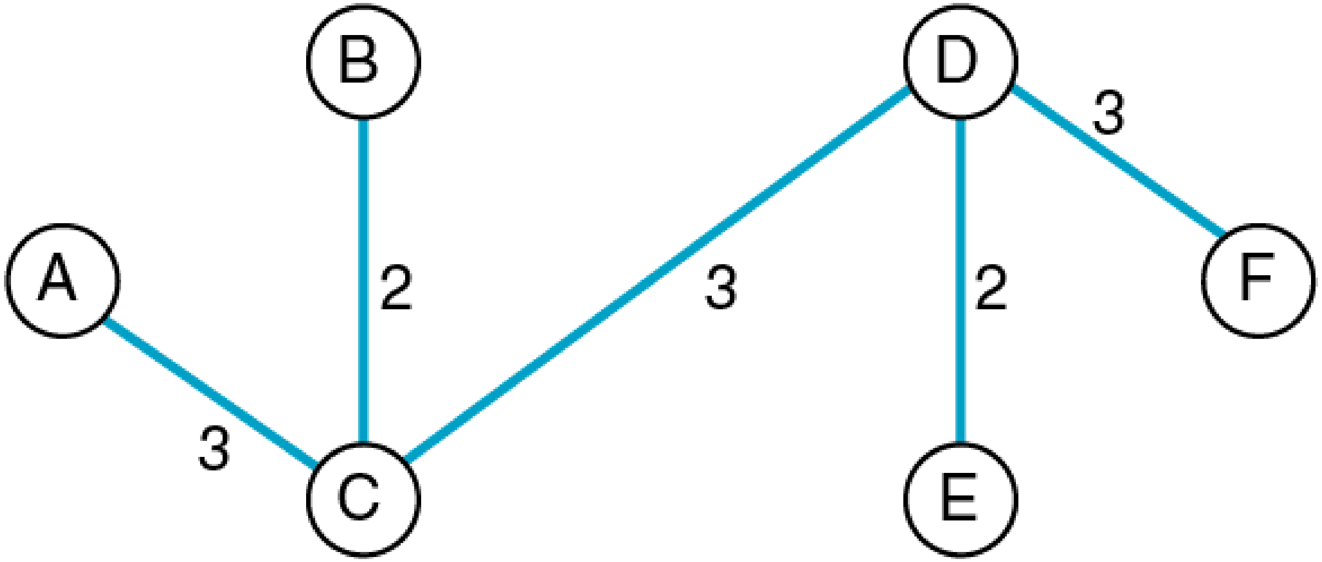




(e) Insert edge DE

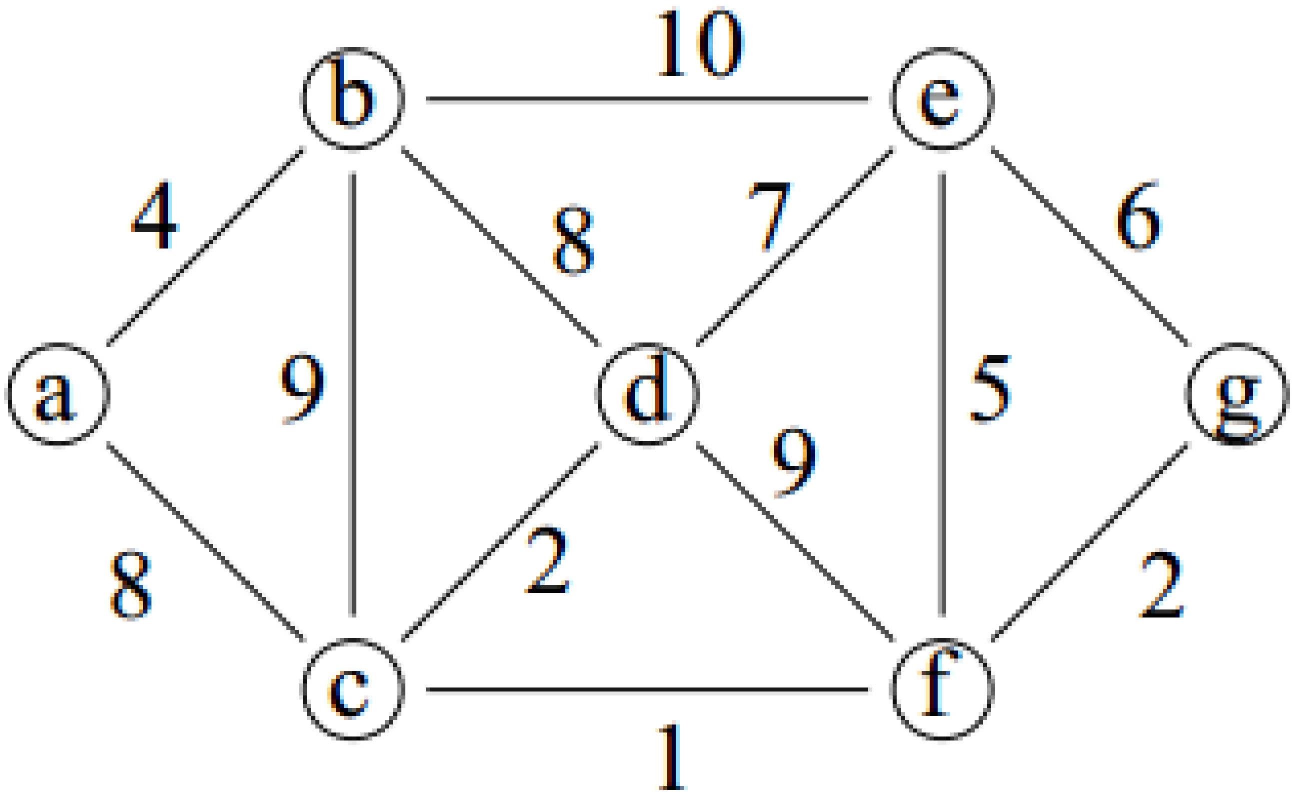
Prim’s Algorithm

- Here is our final MST.



The MST produced by Prim’s algorithm may vary depending on which vertex you start at.

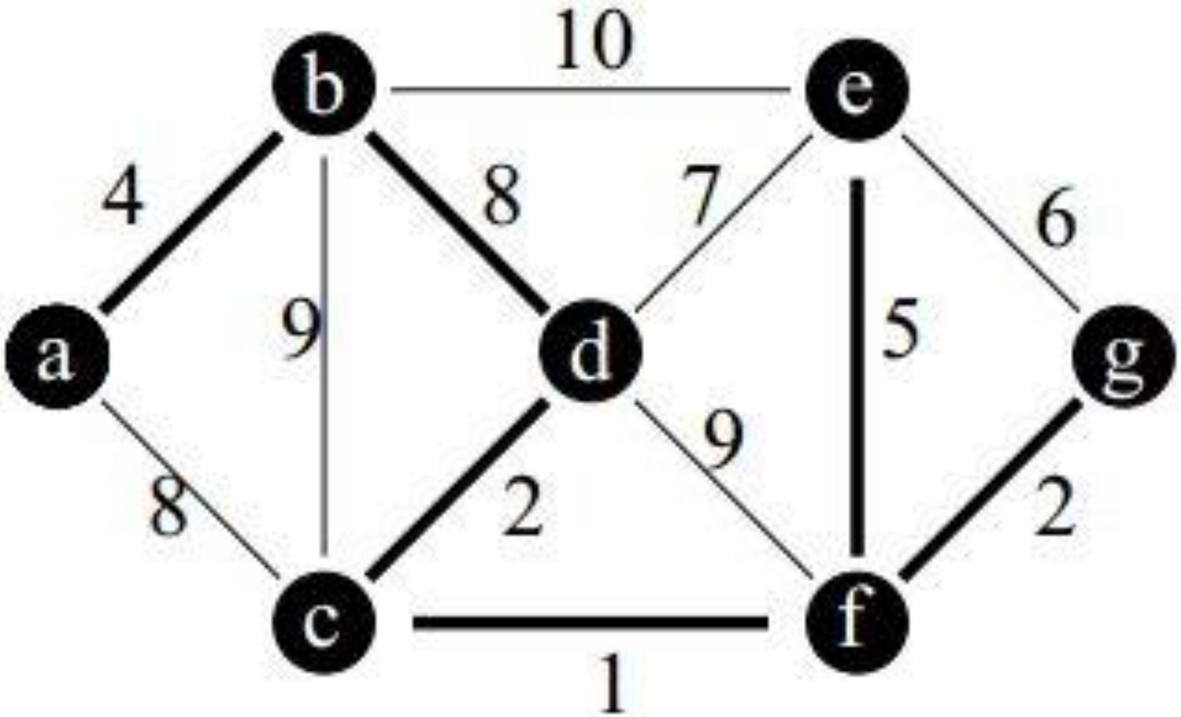
MST Example for Practice (Prim’s)



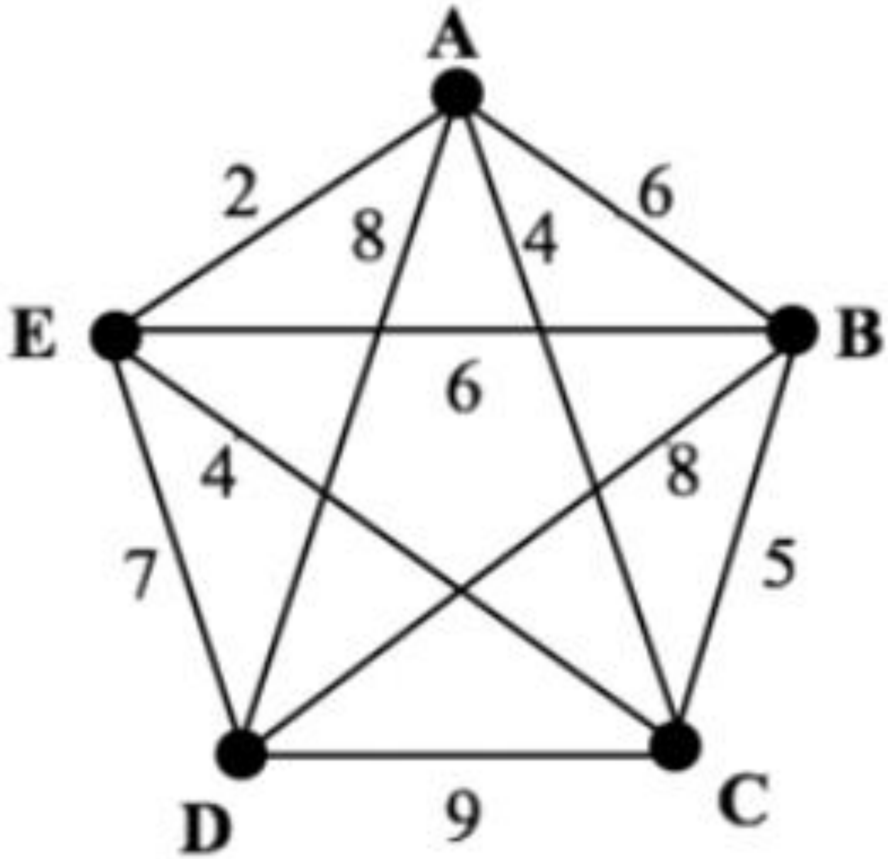
Minimum Total weight: 22

MST Example for Practice (Prim's)

Solution



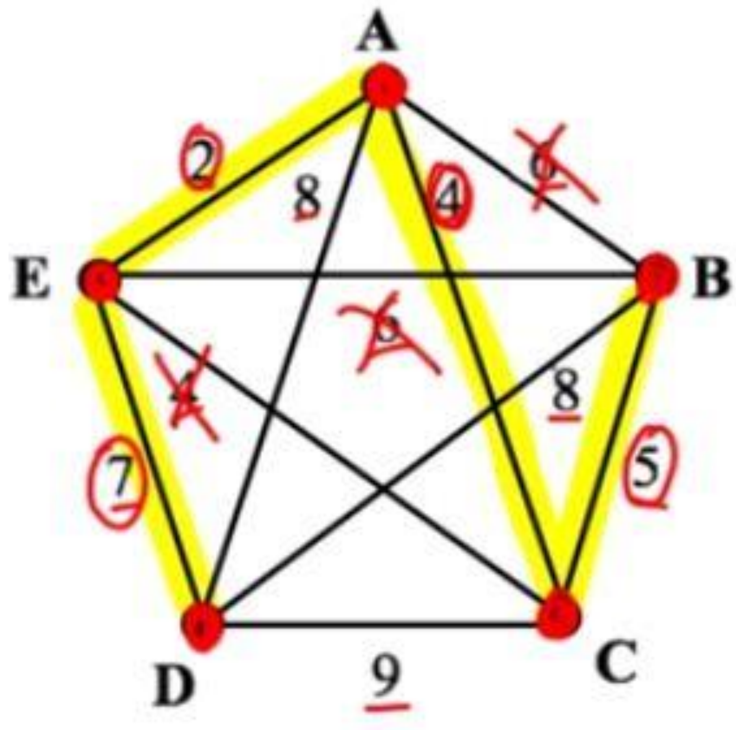
MST Example for Practice (Prim's)



Minimum Total weight: 18

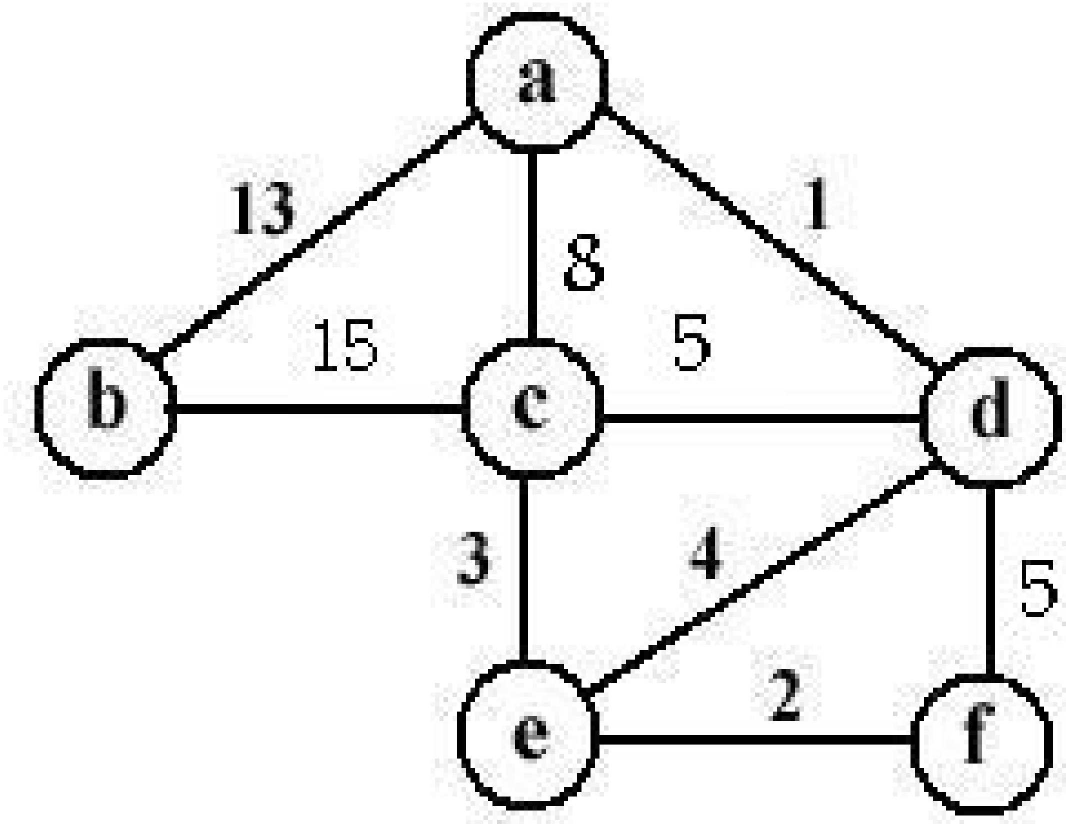
MST Example for Practice (Prim's)

Solution



$7 + 2 + 4 + 5 = 18$

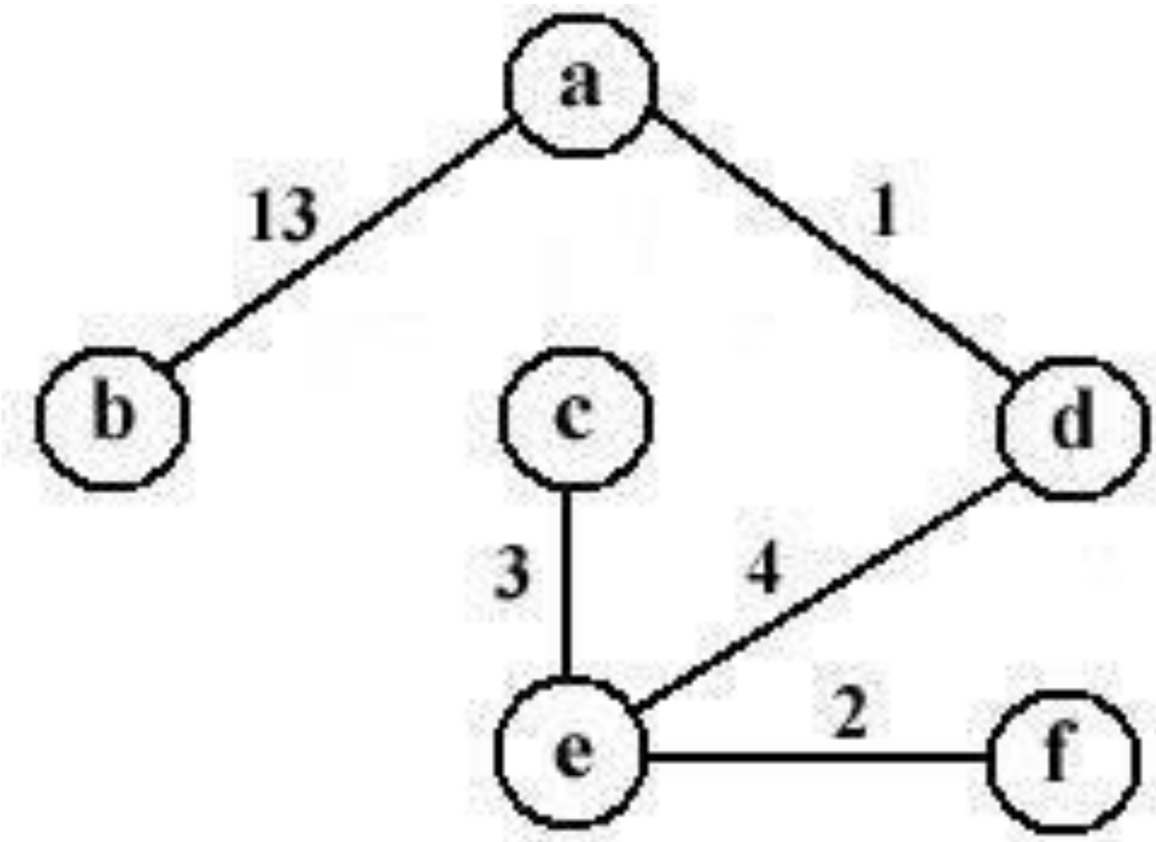
MST Example for Practice (Prim's)



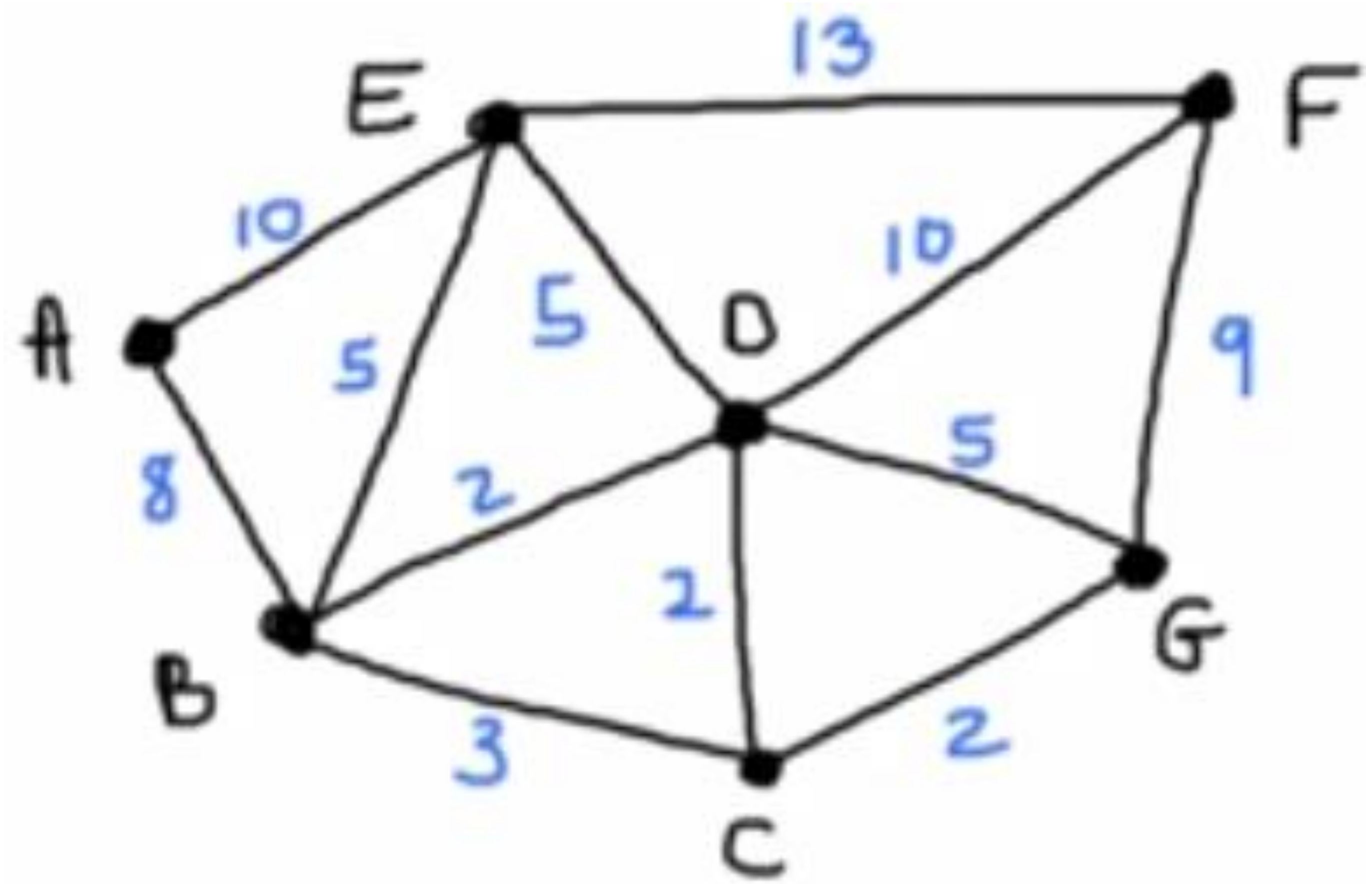
Minimum Total weight: 23

MST Example for Practice (Prim's)

Solution



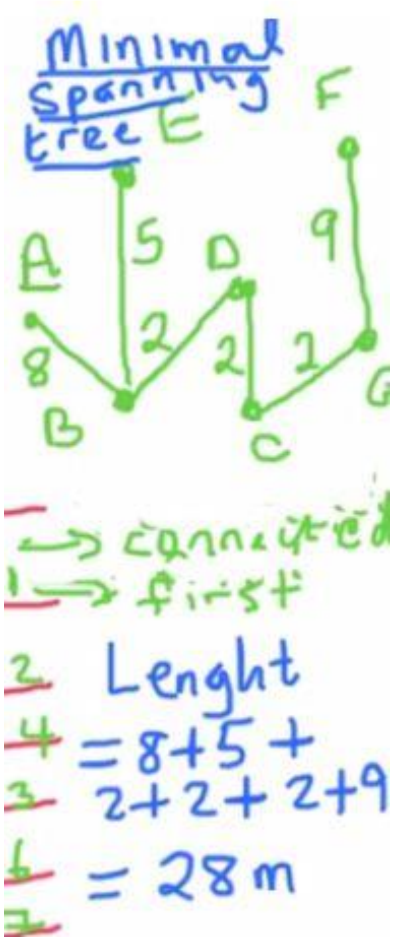
MST Example for Practice (Prim's)



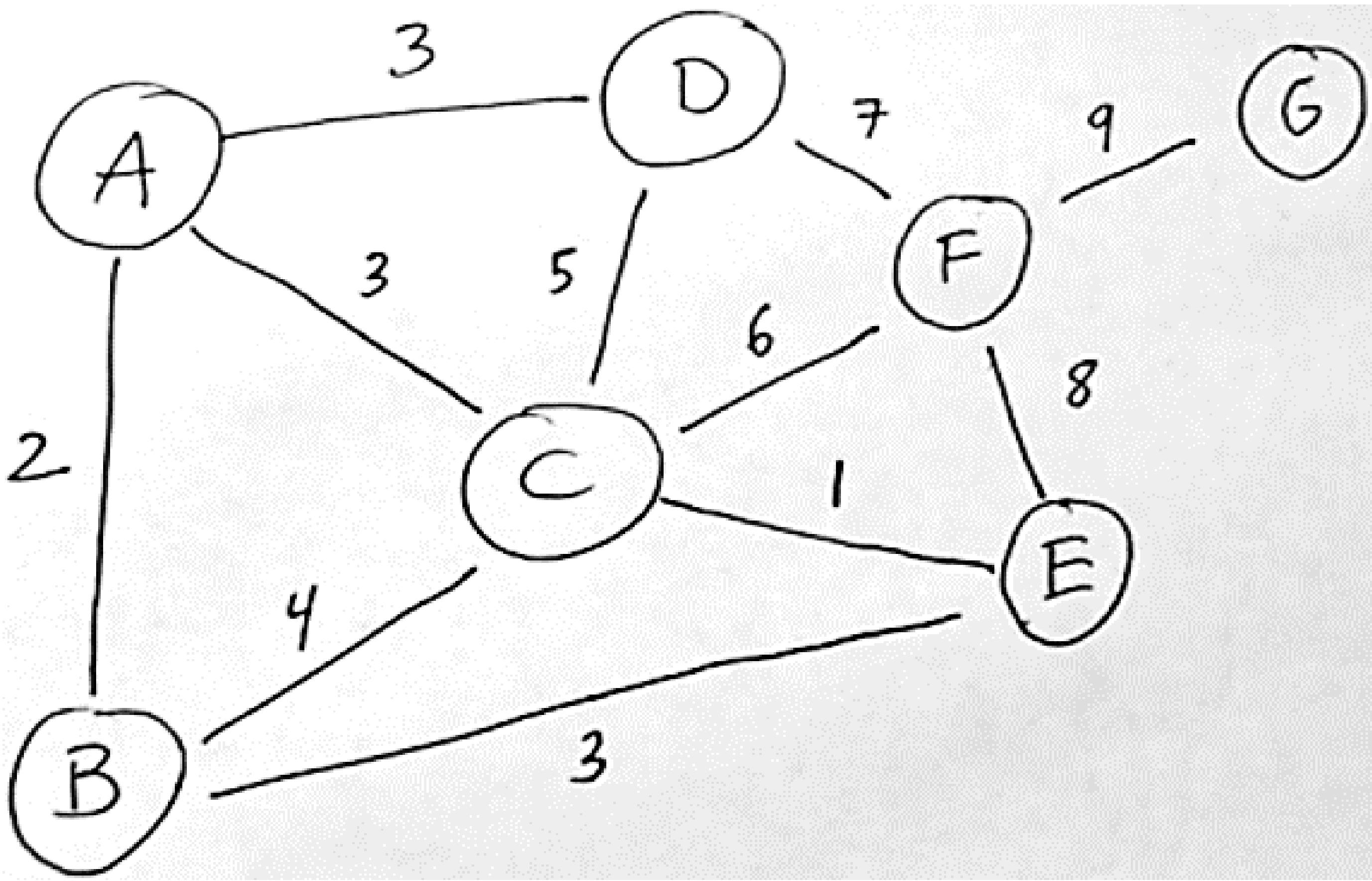
Minimum Total weight: 28

MST Example for Practice (Prim's)

Solution



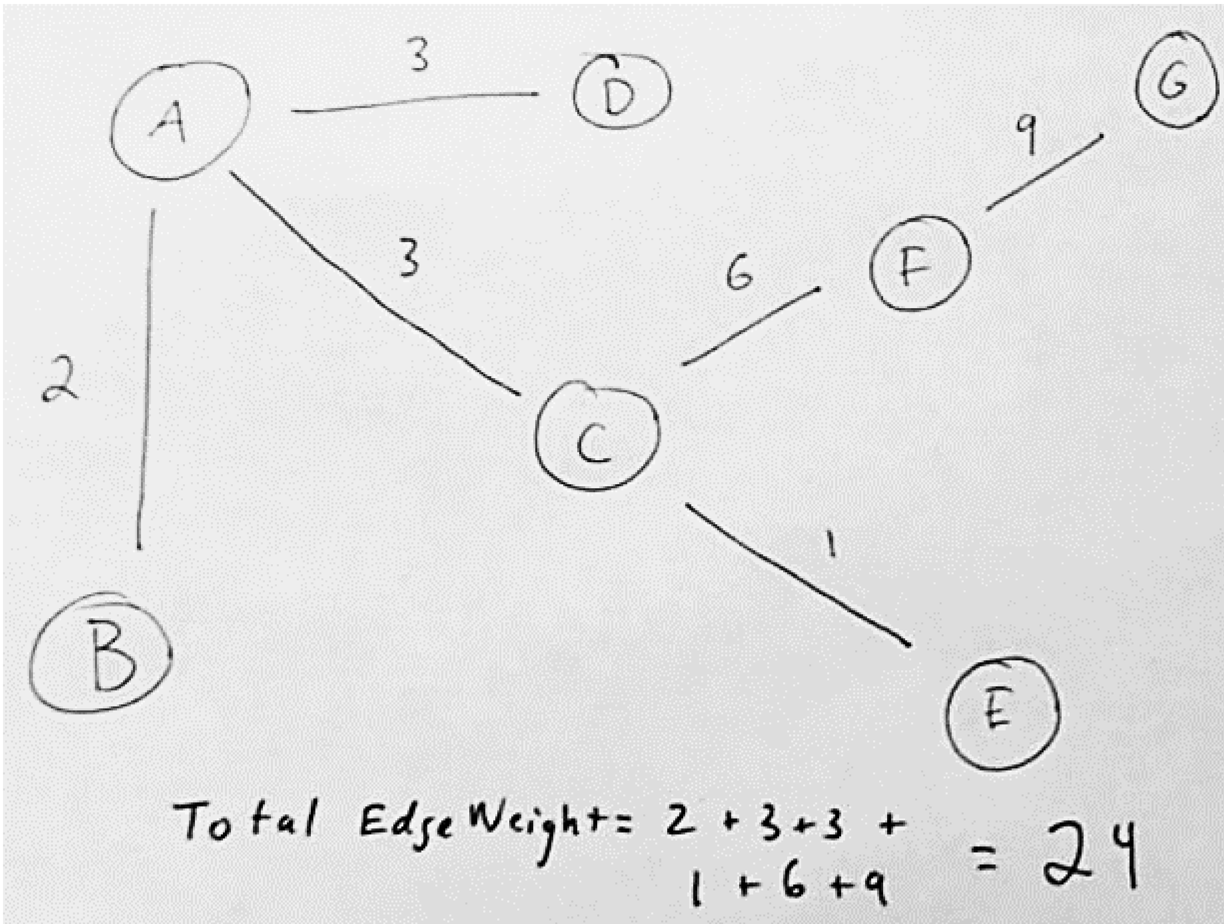
MST Example for Practice (Prim's)



Minimum Total weight: 24

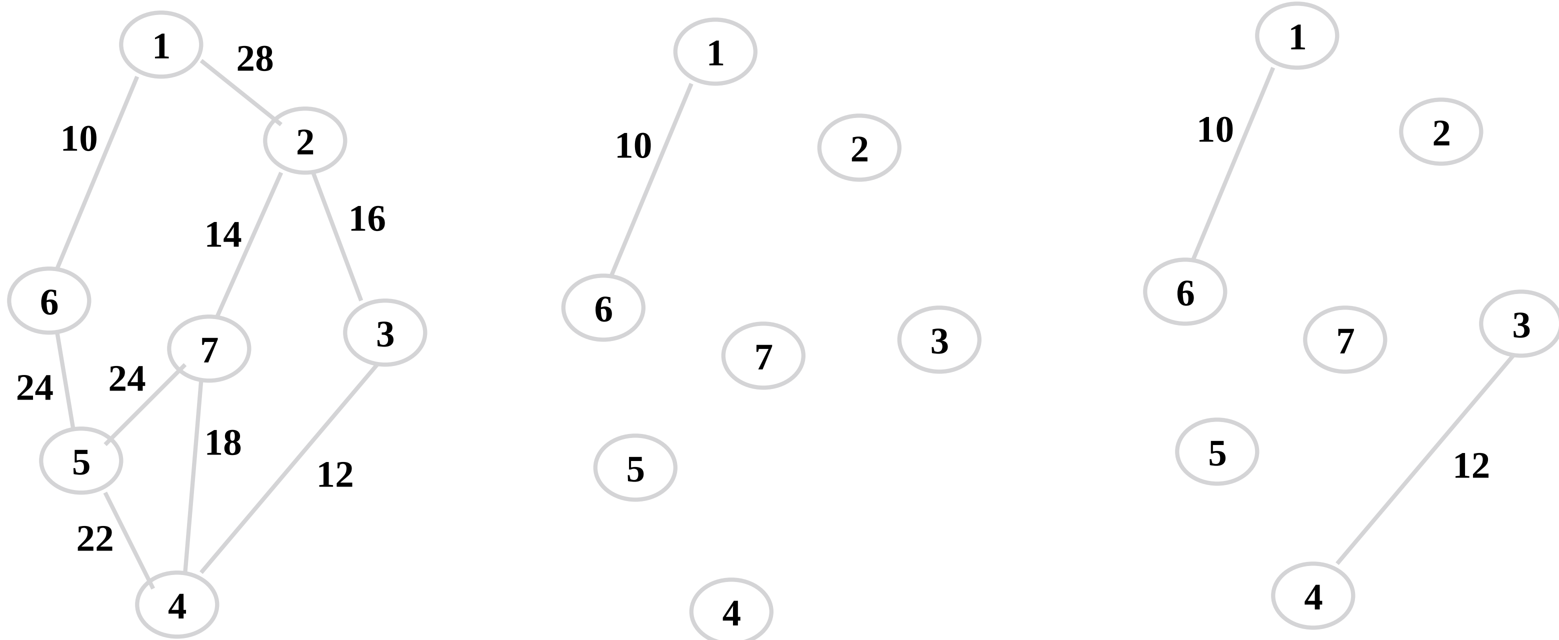
MST Example for Practice (Prim's)

Solution



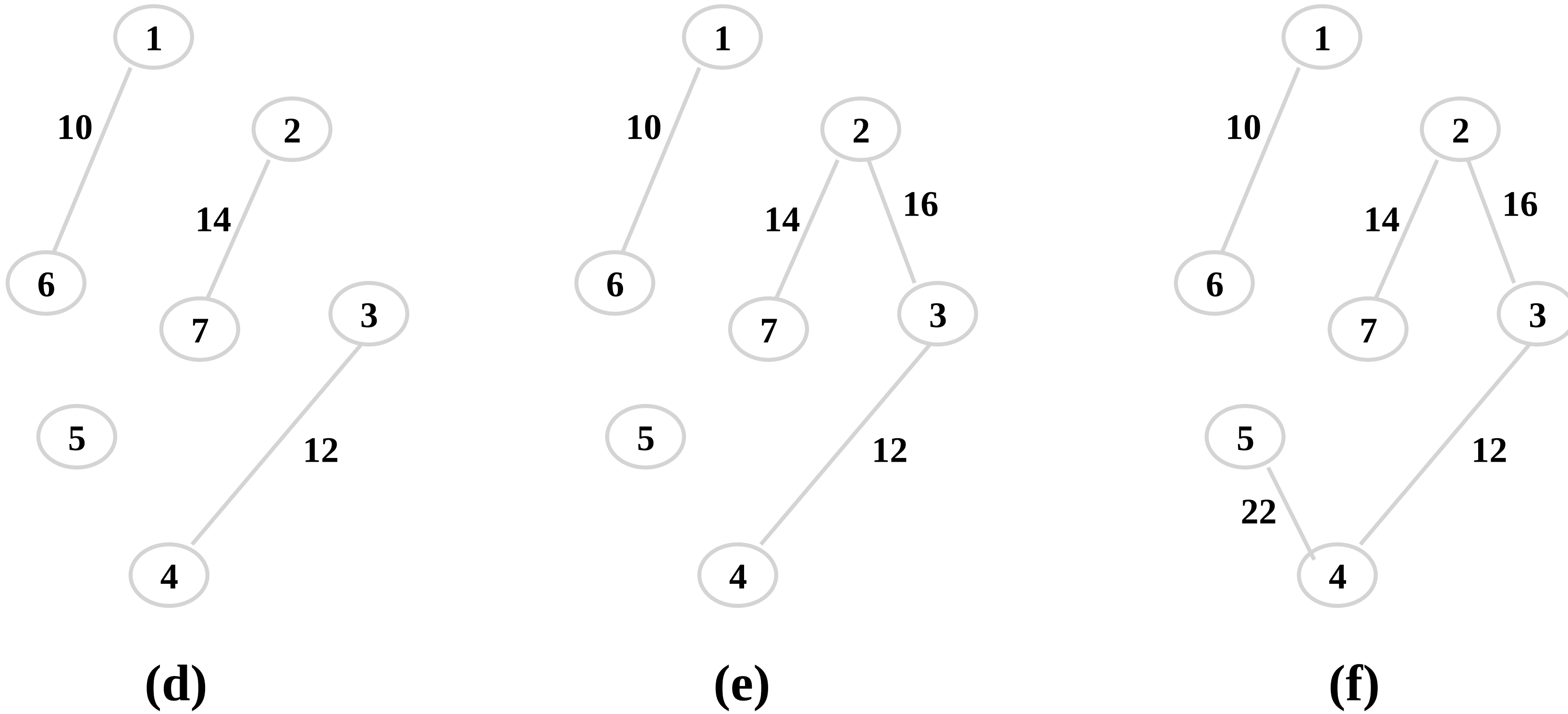
Kruskal’s Algorithm for MST

Kruskal's Algorithm

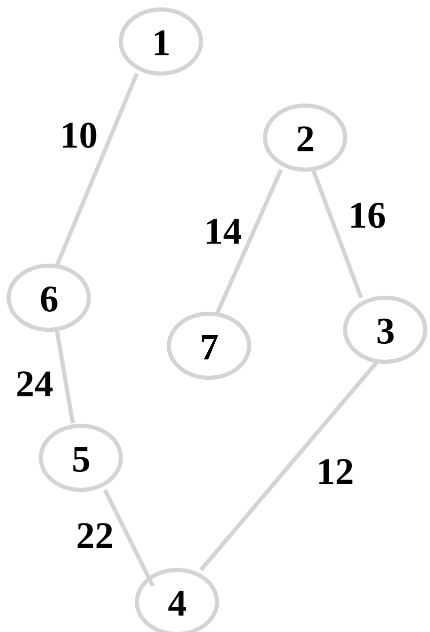


(a) (b) (c)

Kruskal's Algorithm

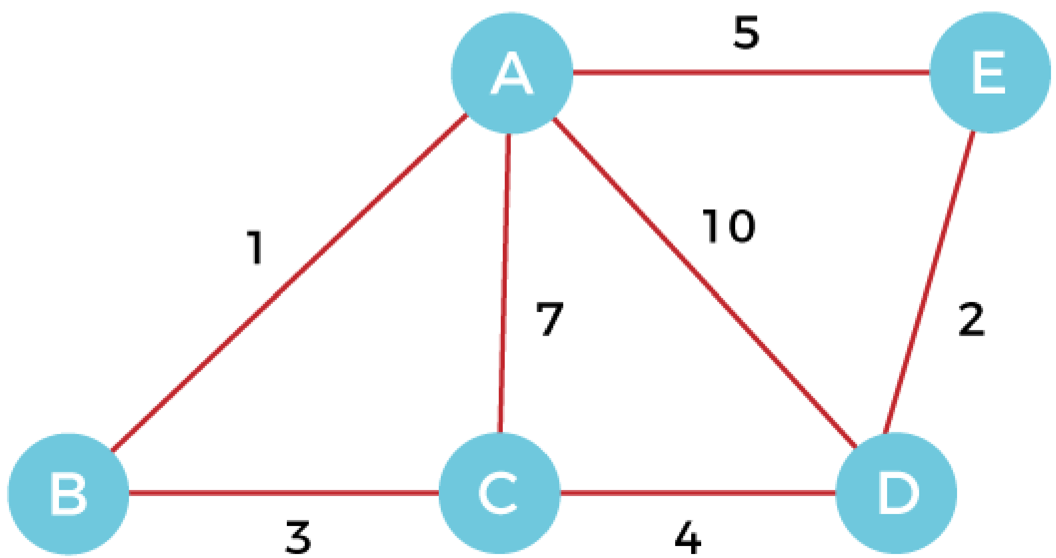


Kruskal's Algorithm

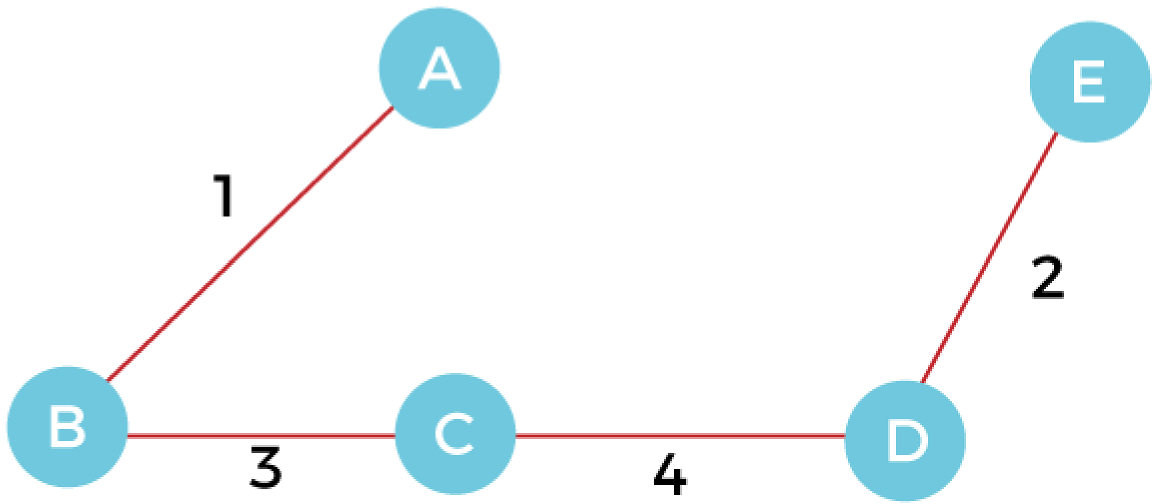


(g)

MST Example for Practice (Kruskal)



MST Example for Practice (Kruskal)

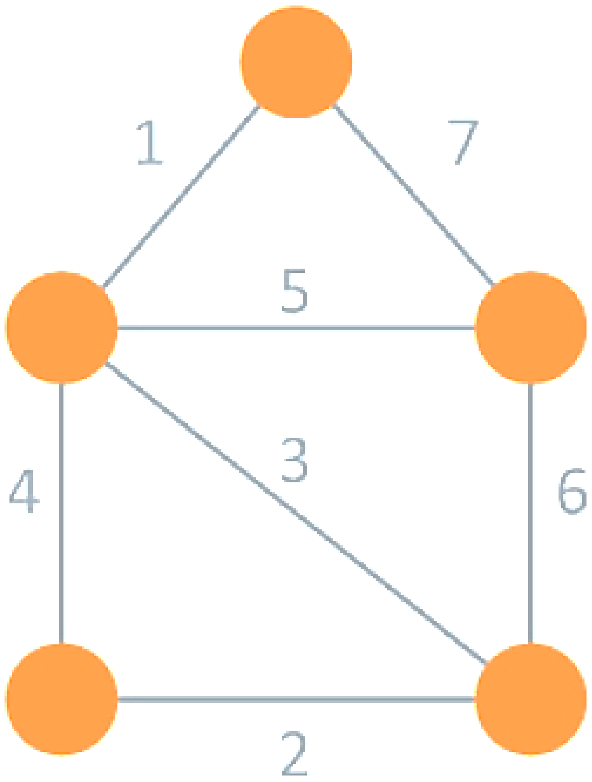


The cost of the MST is = AB + DE + BC + CD

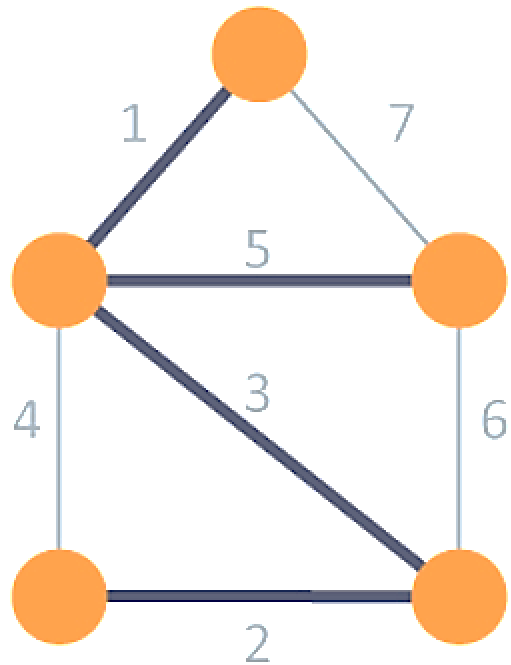
= 1 + 2 + 3 + 4

= 10

MST Example for Practice (Kruskal)

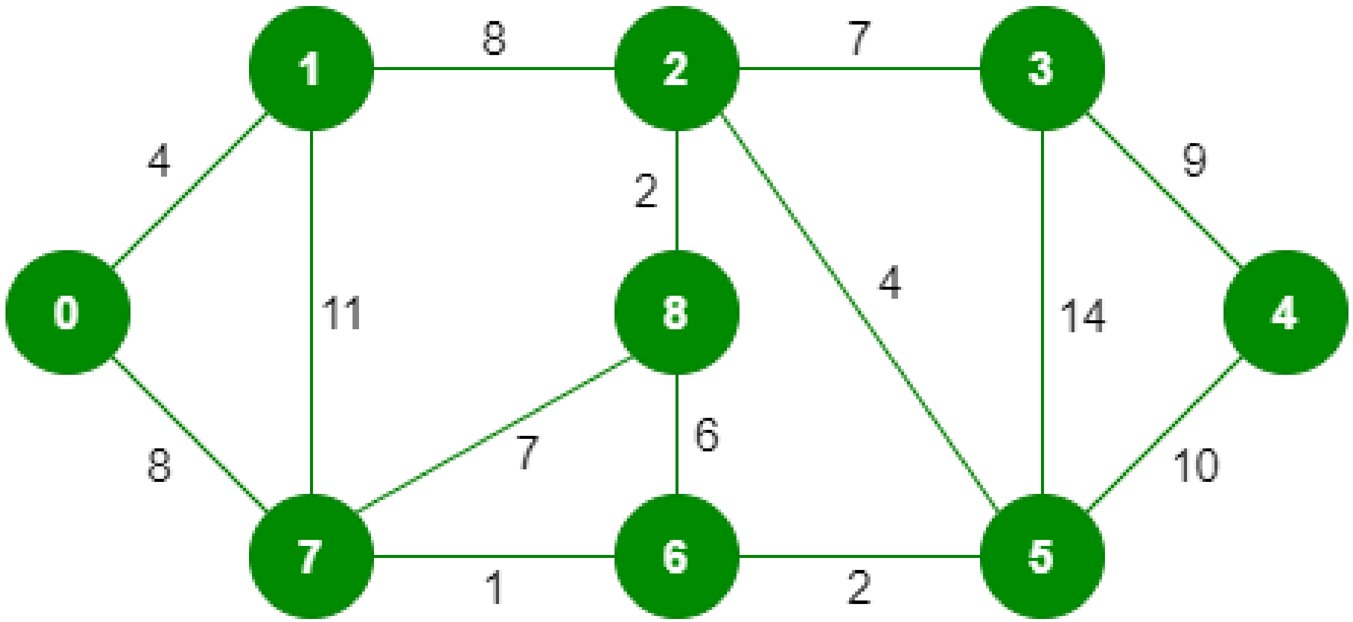


MST Example for Practice (Kruskal)

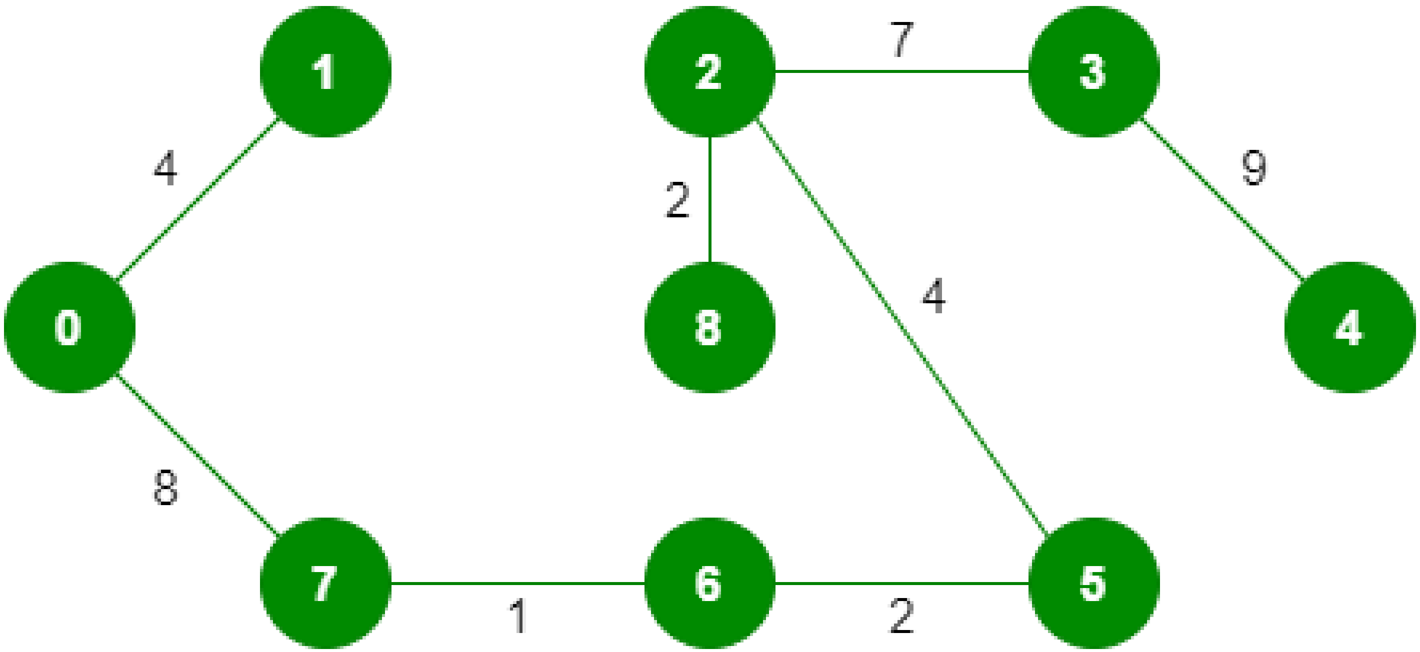


Minimum Total weight: 11

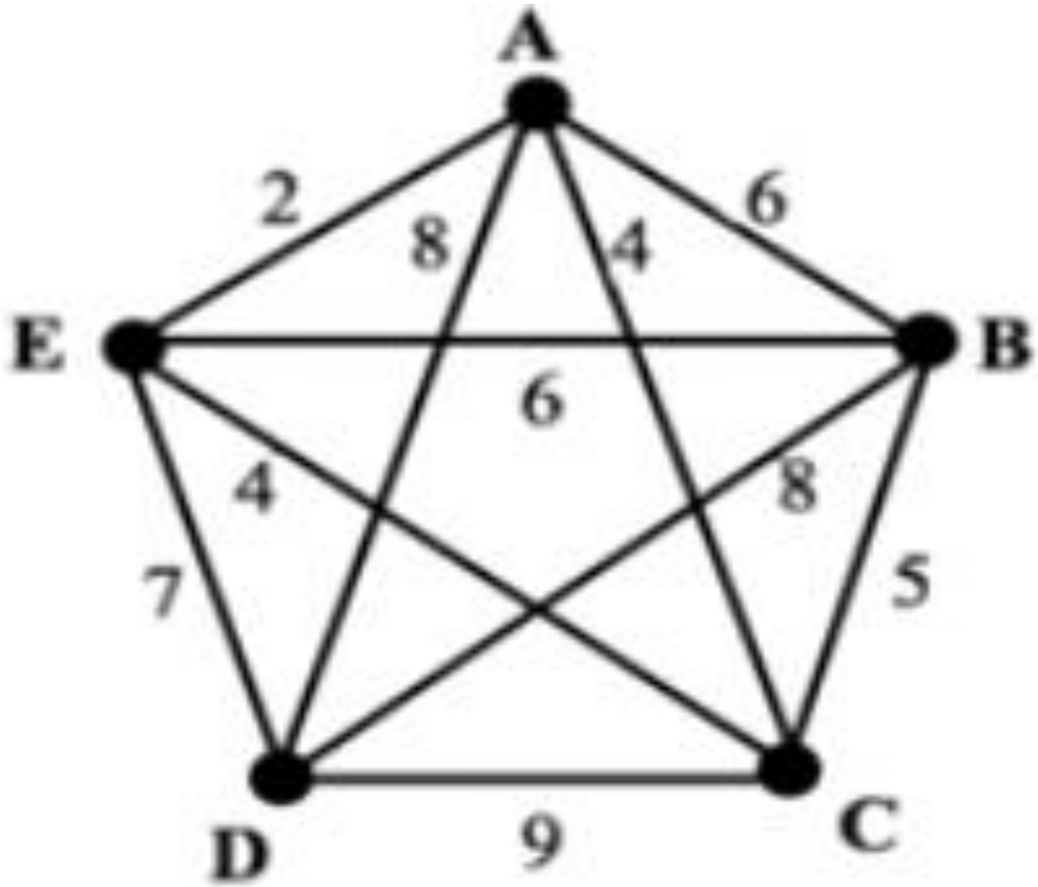
MST Example for Practice (Kruskal)



MST Example for Practice (Kruskal)

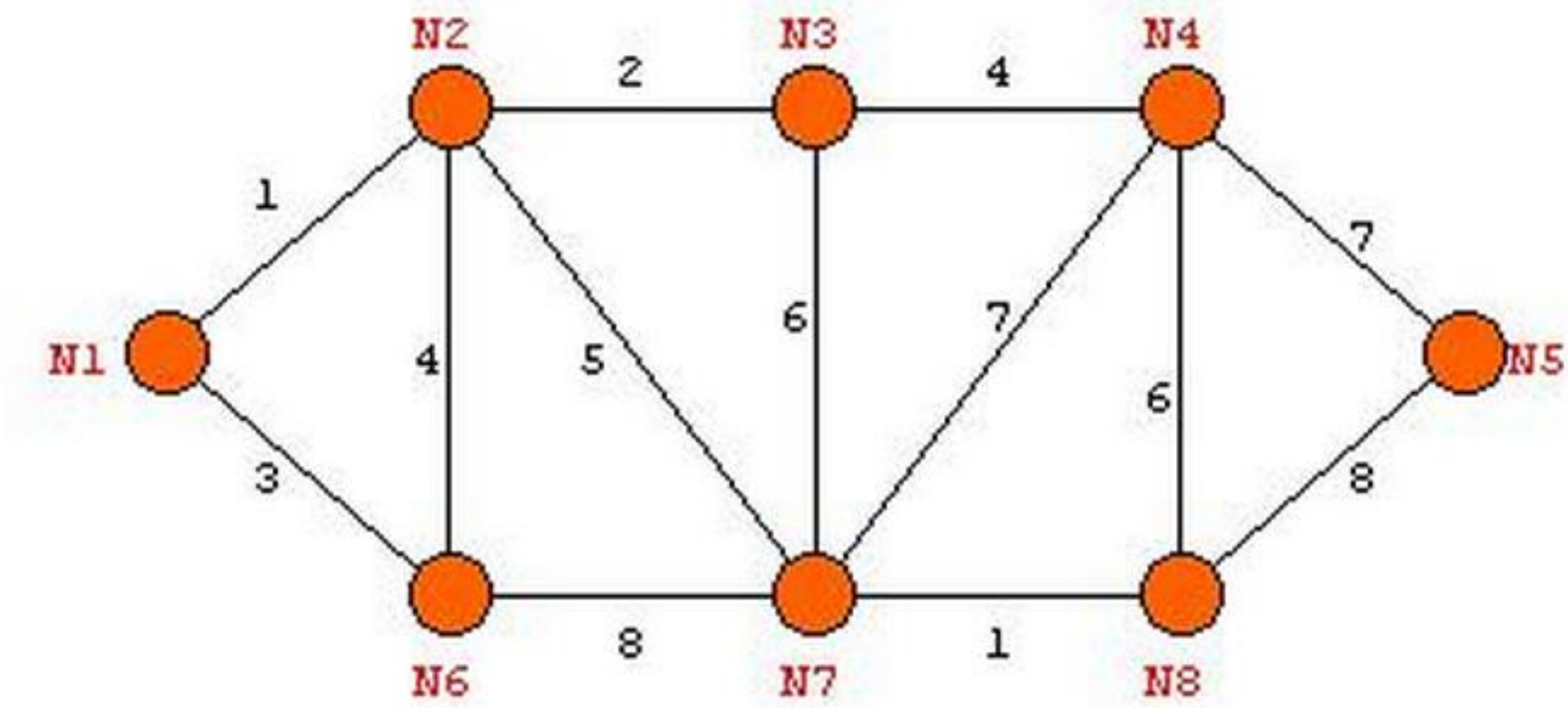


MST Example for Practice (Kruskal)



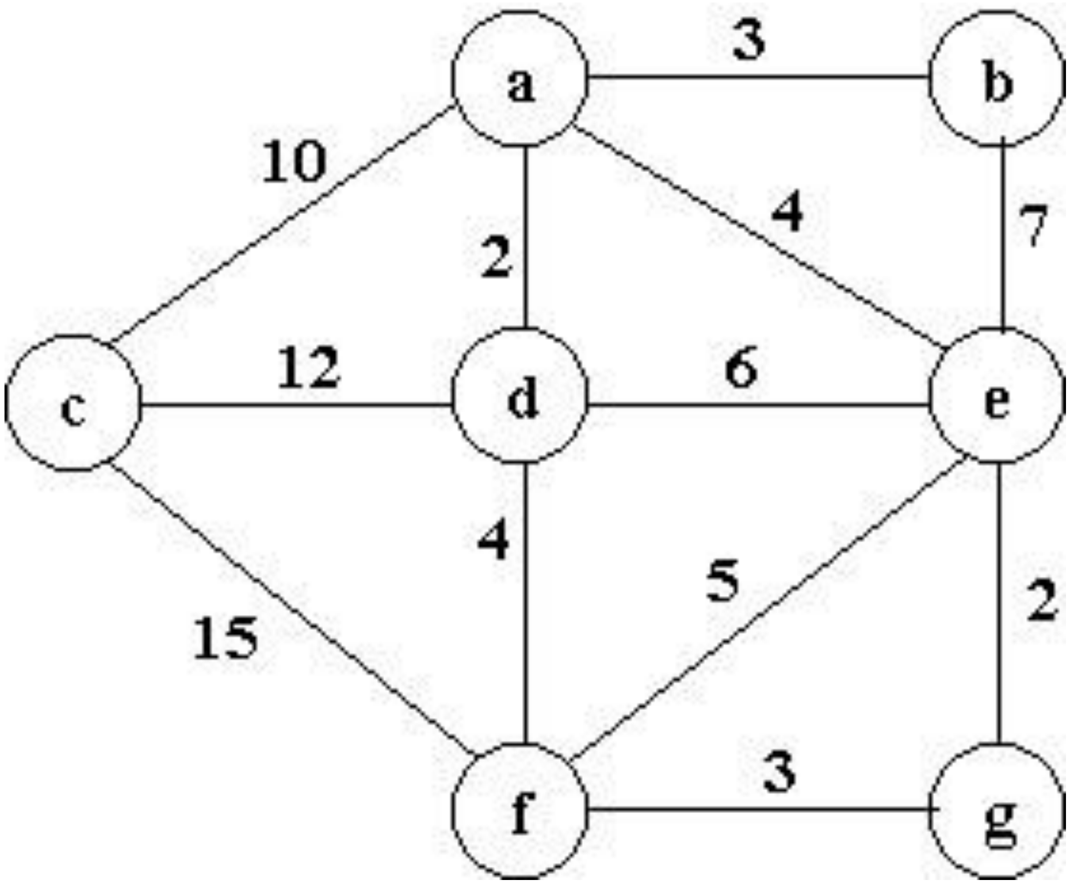
Minimum Total weight: 18

MST Example for Practice (Kruskal)



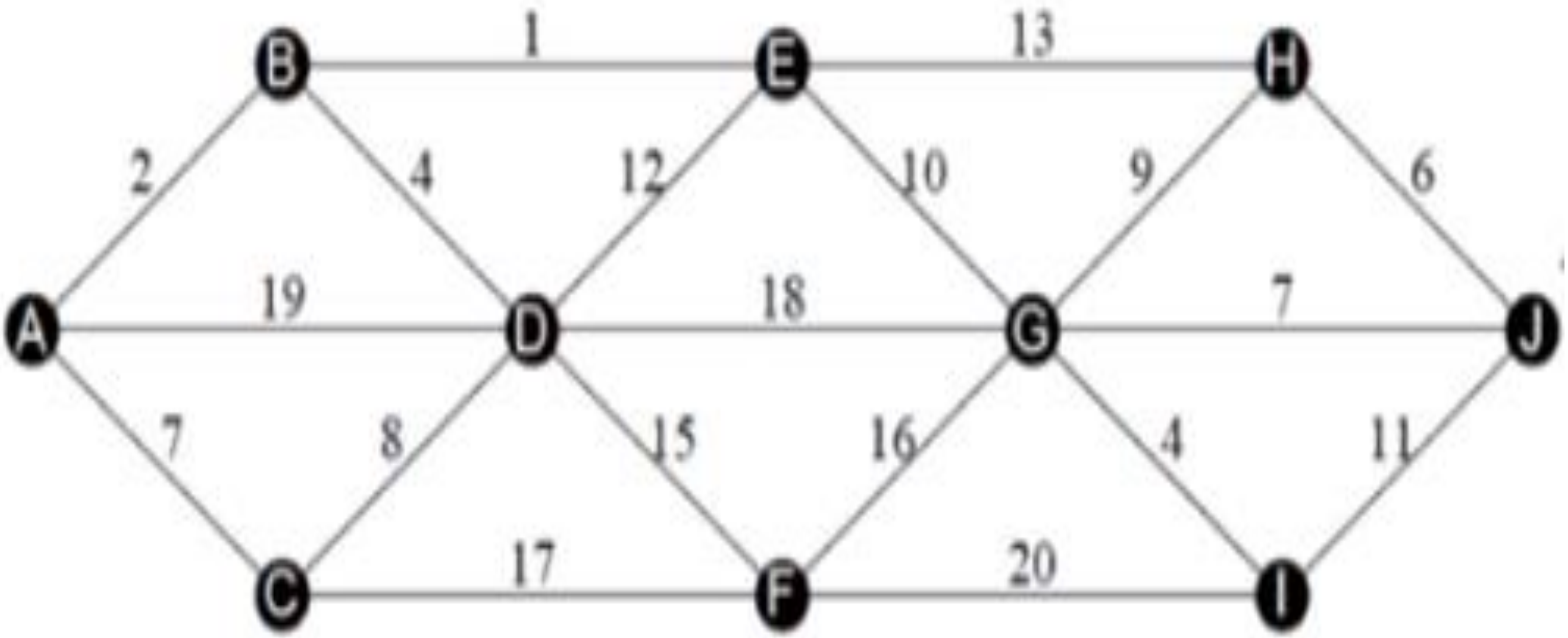
Minimum Total weight: 23

MST Example for Practice (Kruskal)



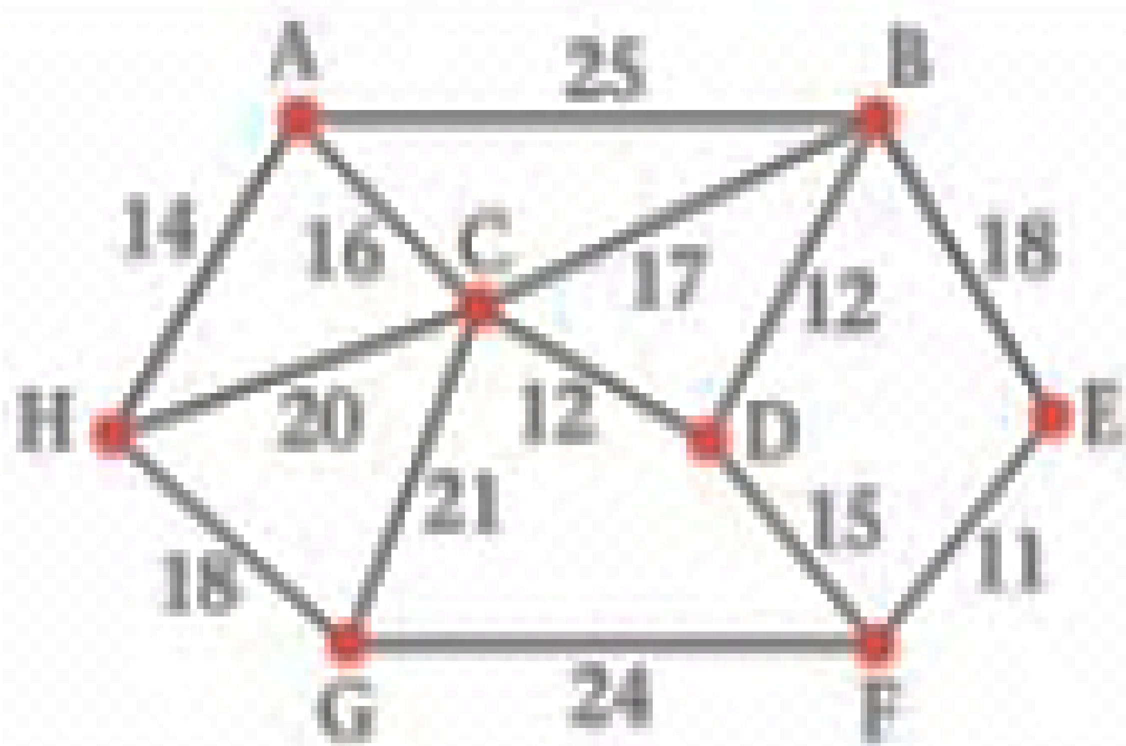
Minimum Total weight: 24

MST Example for Practice (Kruskal)



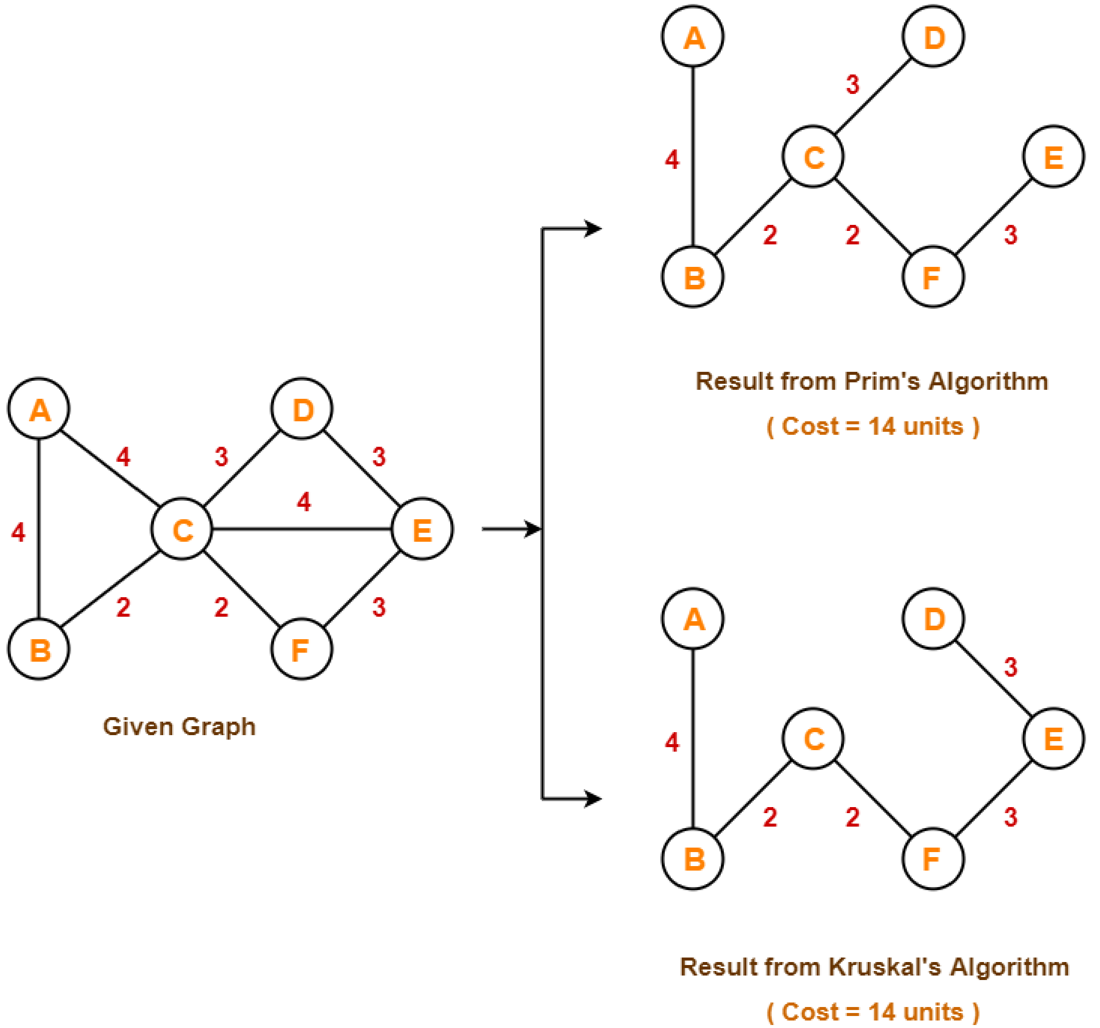
Minimum Total weight: 53

MST Example for Practice (Kruskal)



Minimum Total weight: 98

MST Example for Practice (Kruskal)




Thank You!