Minimum Spanning Tree

GRAPH THEROY

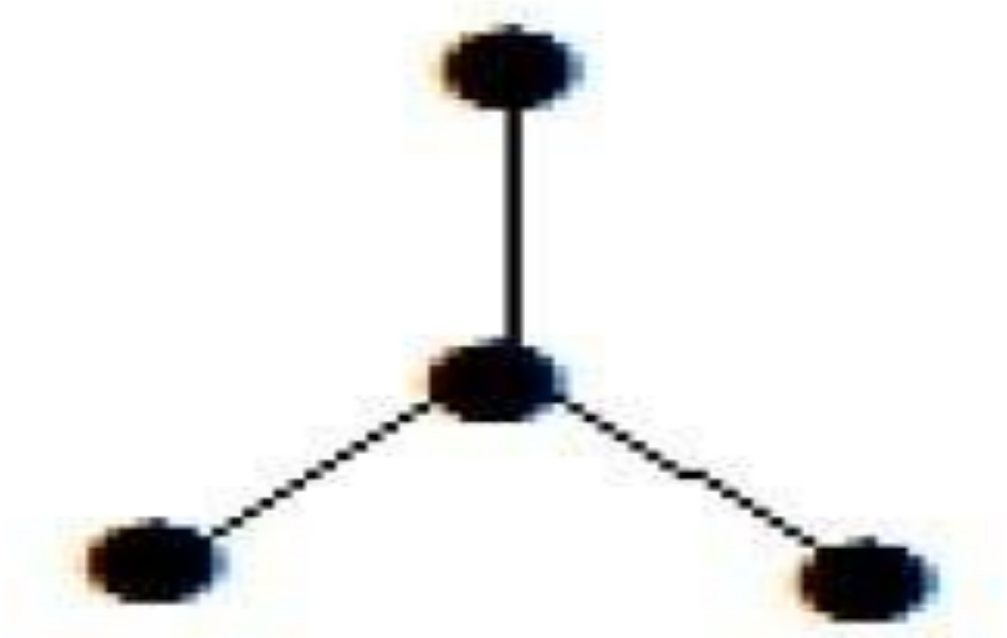
Tree

A **connected acyclic graph** is called a tree. In other words, a connected graph with no cycles is called a tree.

The edges of a tree are known as **branches**. Elements of trees are called their **nodes**. The nodes without child nodes are called **leaf nodes**.

A tree with ‘n’ vertices has ‘n-1’ edges.

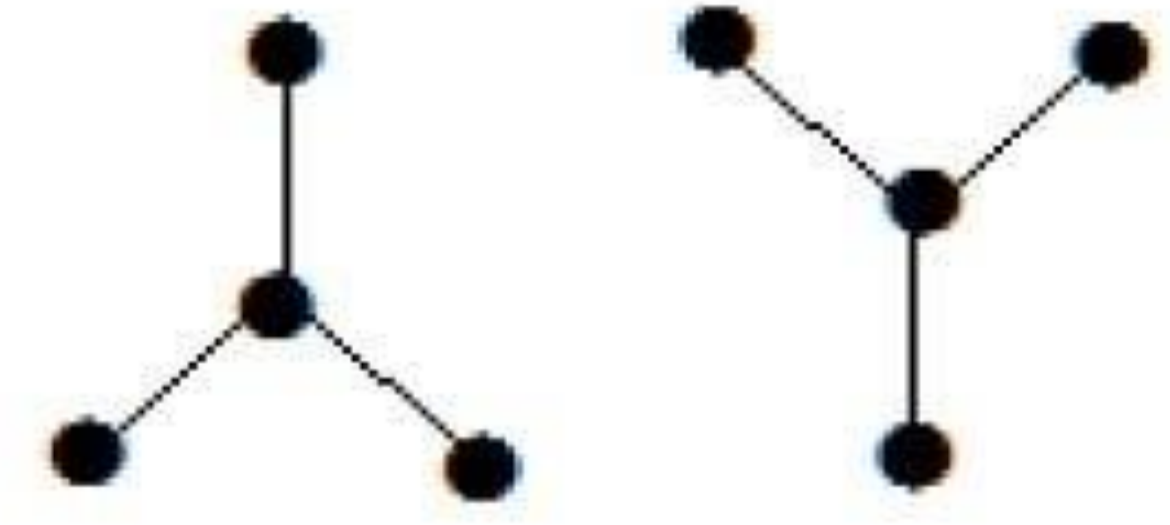
Example : The graph shown here is a tree because it has no cycles and it is connected. It has four vertices and three edges, i.e., for ‘n’ vertices ‘n-1’ edges as mentioned in the definition.



Forest

A **disconnected acyclic graph** is called a forest. In other words, a disjoint collection of trees is called a forest.

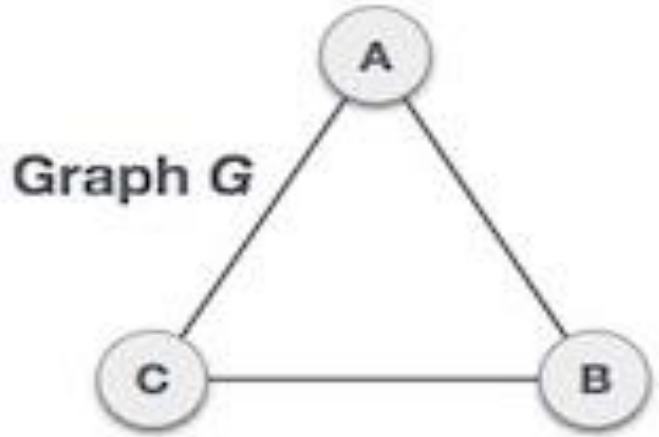
Example

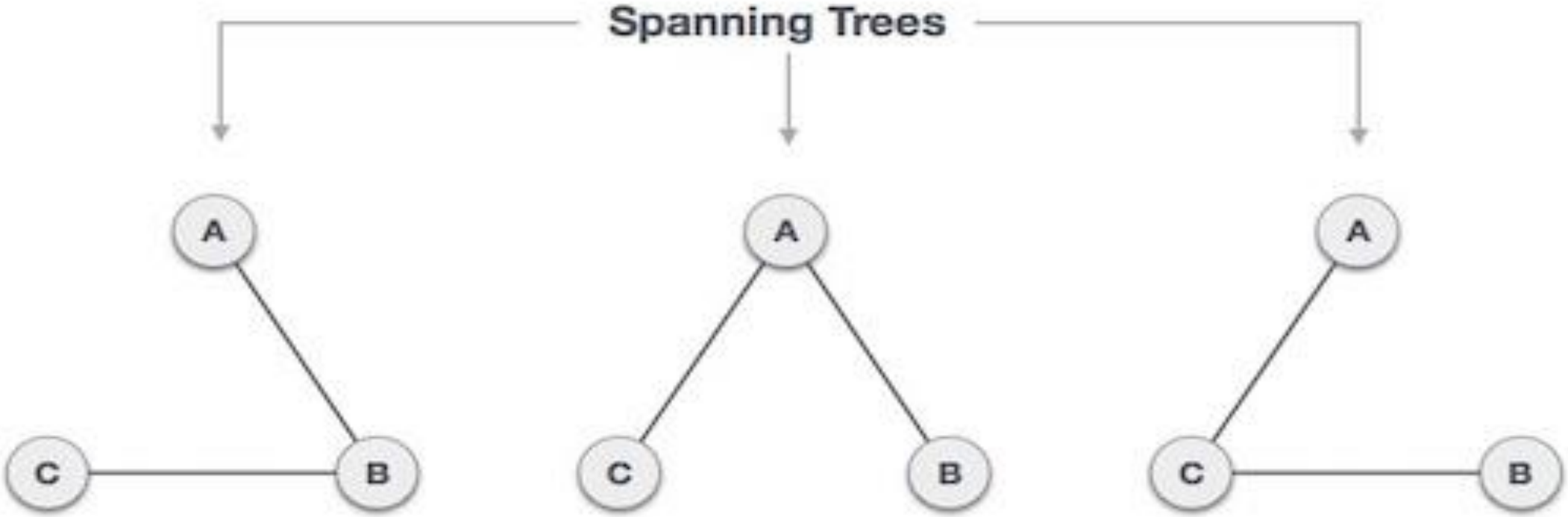


Spanning Tree

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.


Graph G
A
C
B


Spanning Trees
A
C
B
A
C
B
A
C
B

A connected graph G can have more than one spanning tree.

All possible spanning trees of graph G, have the same number of edges and vertices.

The spanning tree does not have any cycle (loops).

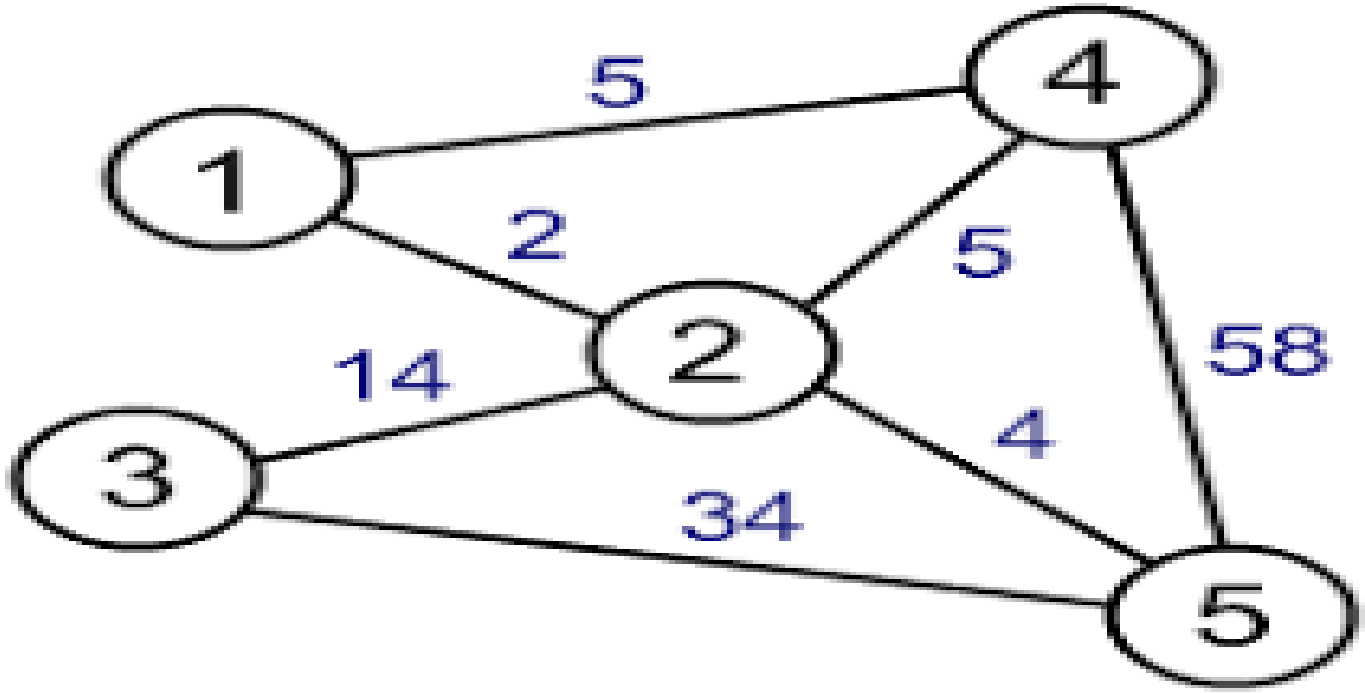
Removing one edge from the spanning tree will make the graph disconnected, i.e. the spanning tree is **minimally connected**.

Adding one edge to the spanning tree will create a circuit or loop, i.e. the spanning tree is **maximally acyclic**.

Weighted Graphs

A weighted graph is a graph in which each branch is given a numerical weight. A weighted graph is therefore a special type of labeled graph in which the labels are numbers (which are usually taken to be positive).

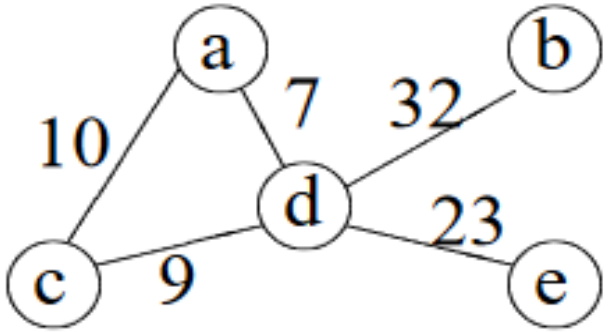
Example:



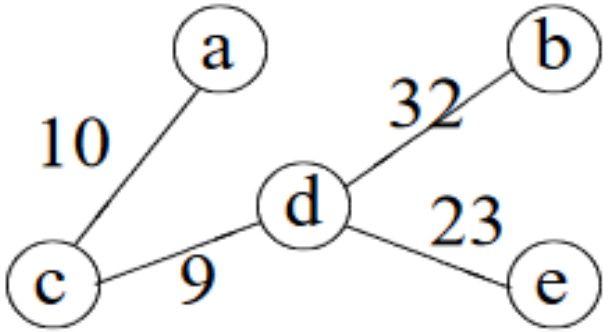
Minimum Spanning Tree

A *minimum spanning tree* in a connected weighted graph is a spanning tree that has the smallest possible sum of weights of its edges.

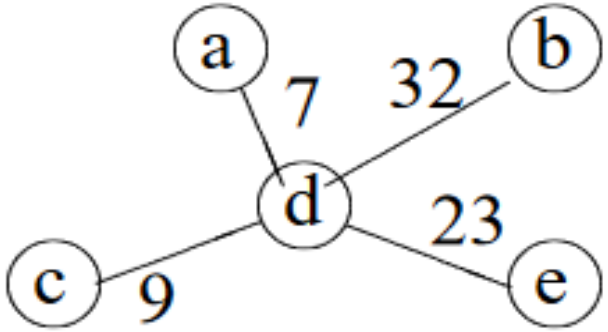
Example: Tree T2 is MST



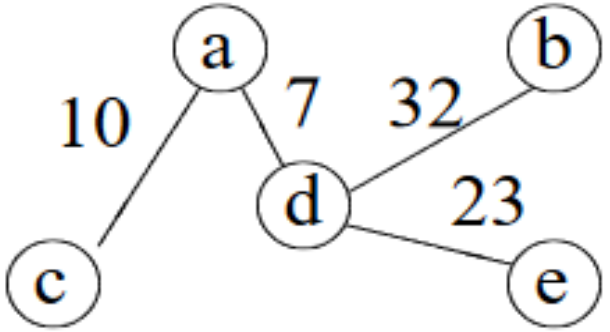
weighted graph



Tree 1. w=74



Tree 2, w=71



Tree 3, w=72

Minimum spanning tree

Kruskal's algorithm

sort the edges of G in increasing order by length

    keep a subgraph S of G, initially empty

    for each edge e in sorted order

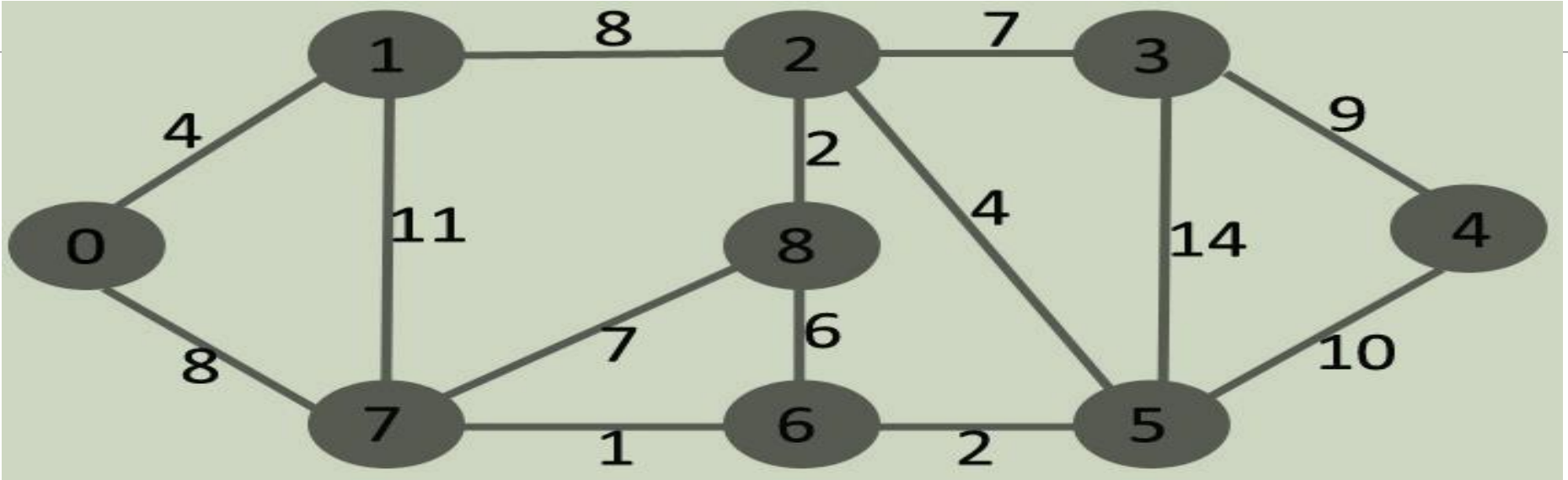
        if the endpoints of e are disconnected in S

        add e to S

    return S

Example: Kruskal's Algorithm

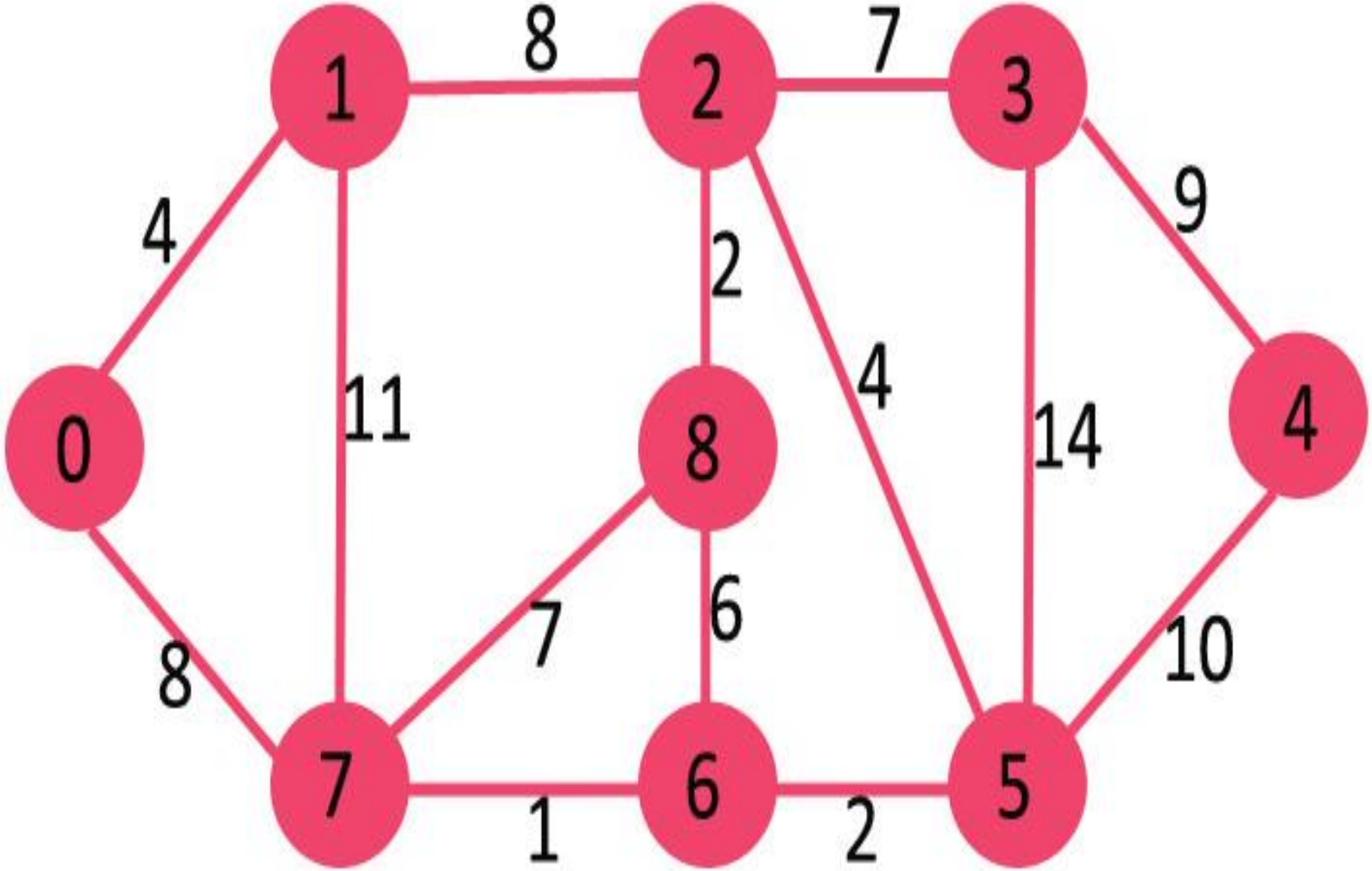
Consider the below input graph.



The graph contains 9 vertices and 14 edges. So, the minimum spanning tree formed will be having (9 – 1) = 8 edges.

After sorting:

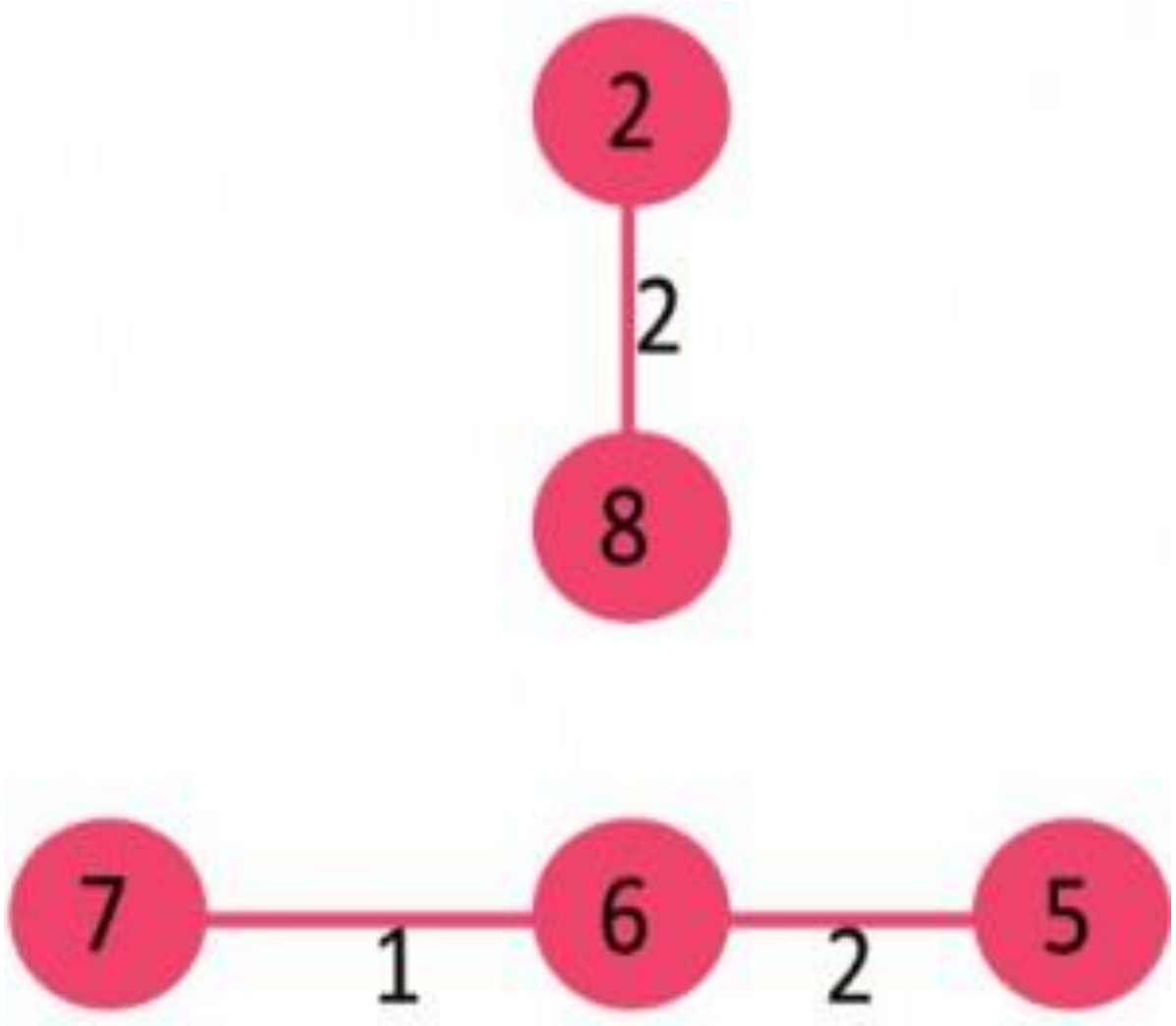
Weight	Src	Dest
1	7	6
2	8	2
2	6	5
4	0	1
4	2	5
6	8	6
7	2	3
7	7	8
8	0	7
8	1	2
9	3	4
10	5	4



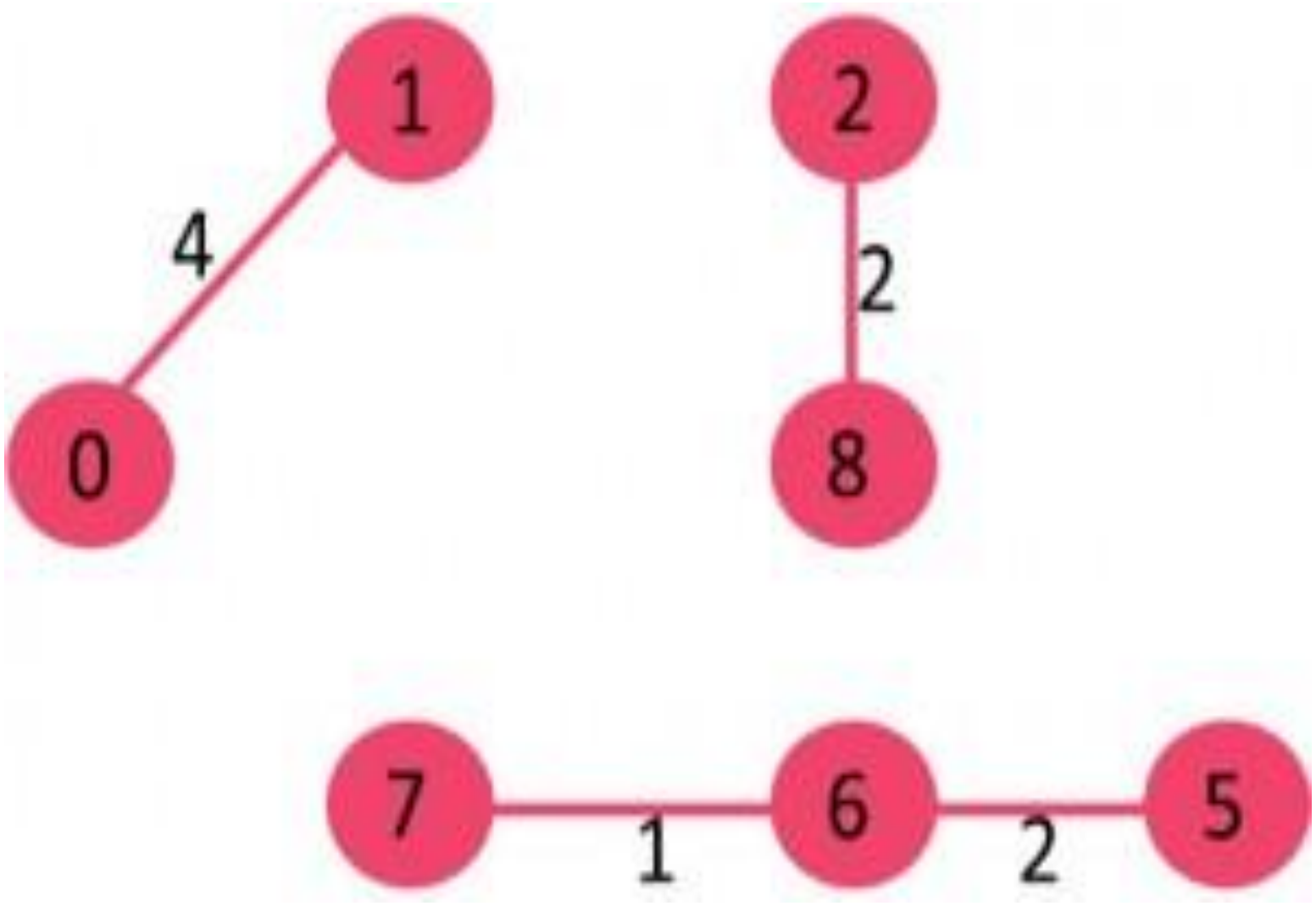
1. Pick edge 7-6: No cycle is formed, include it.

2. Pick edge 8-2: No cycle is formed, include it.

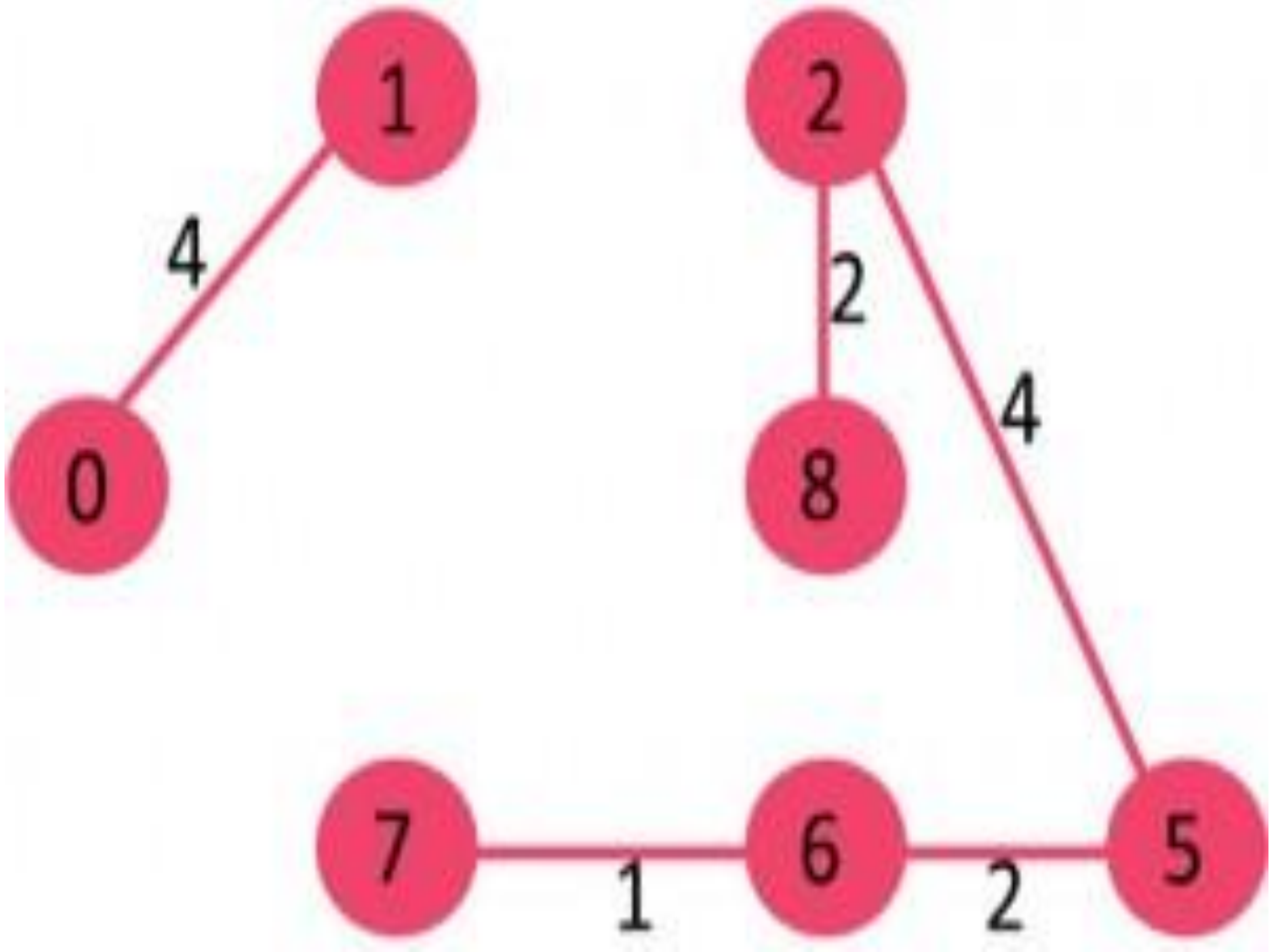
3. Pick edge 6-5: No cycle is formed, include it.



4. *Pick edge 0-1:* No cycle is formed, include it.

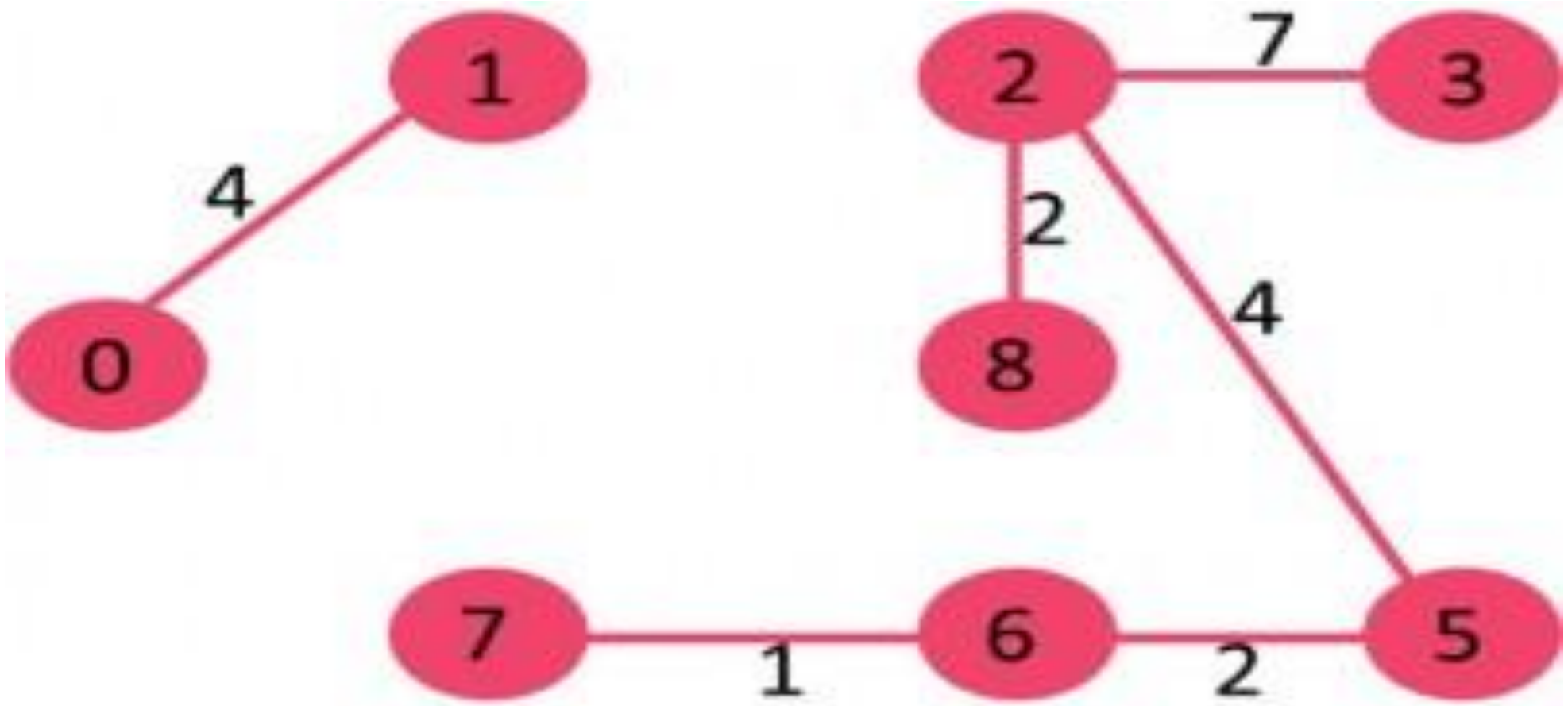


5. *Pick edge 2-5:* No cycle is formed, include it.



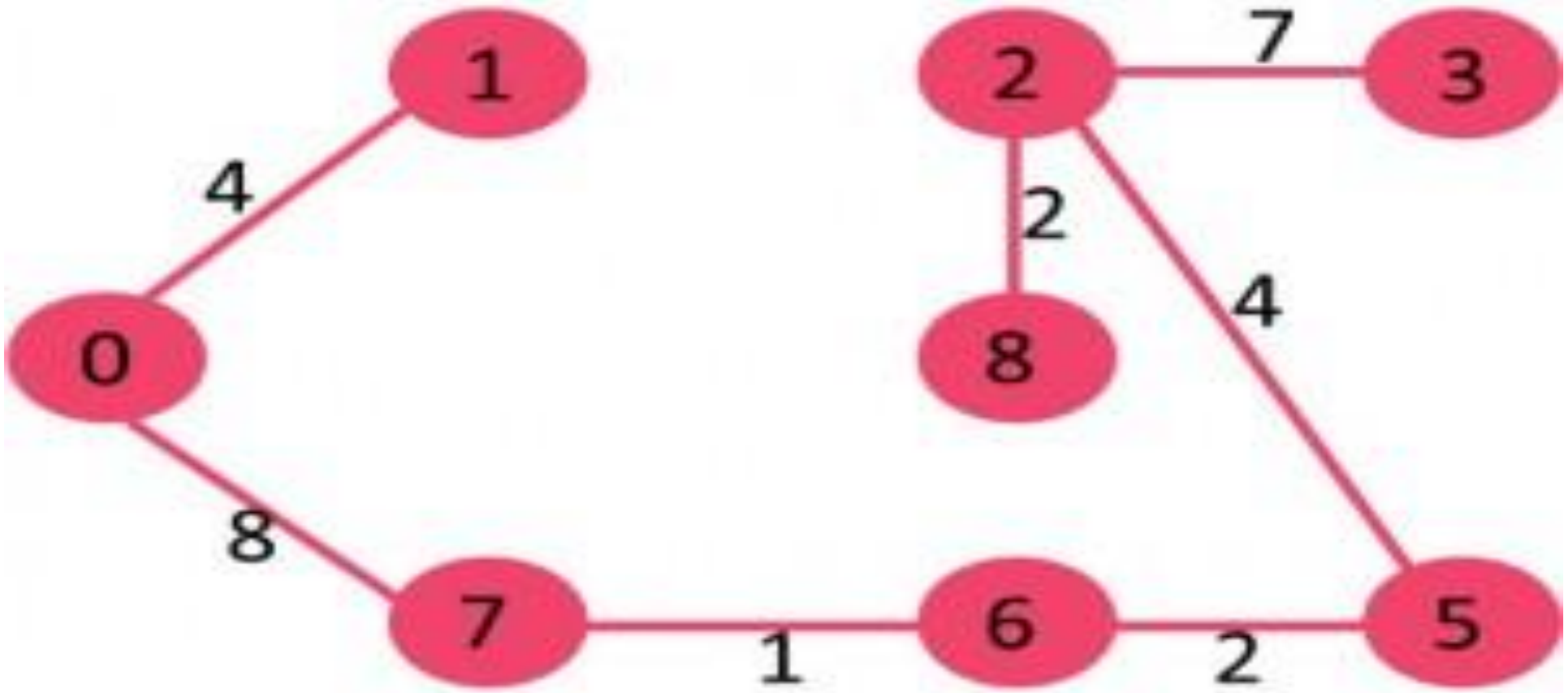
6. *Pick edge 8-6:* Since including this edge results in cycle, discard it.

7. *Pick edge 2-3:* No cycle is formed, include it.



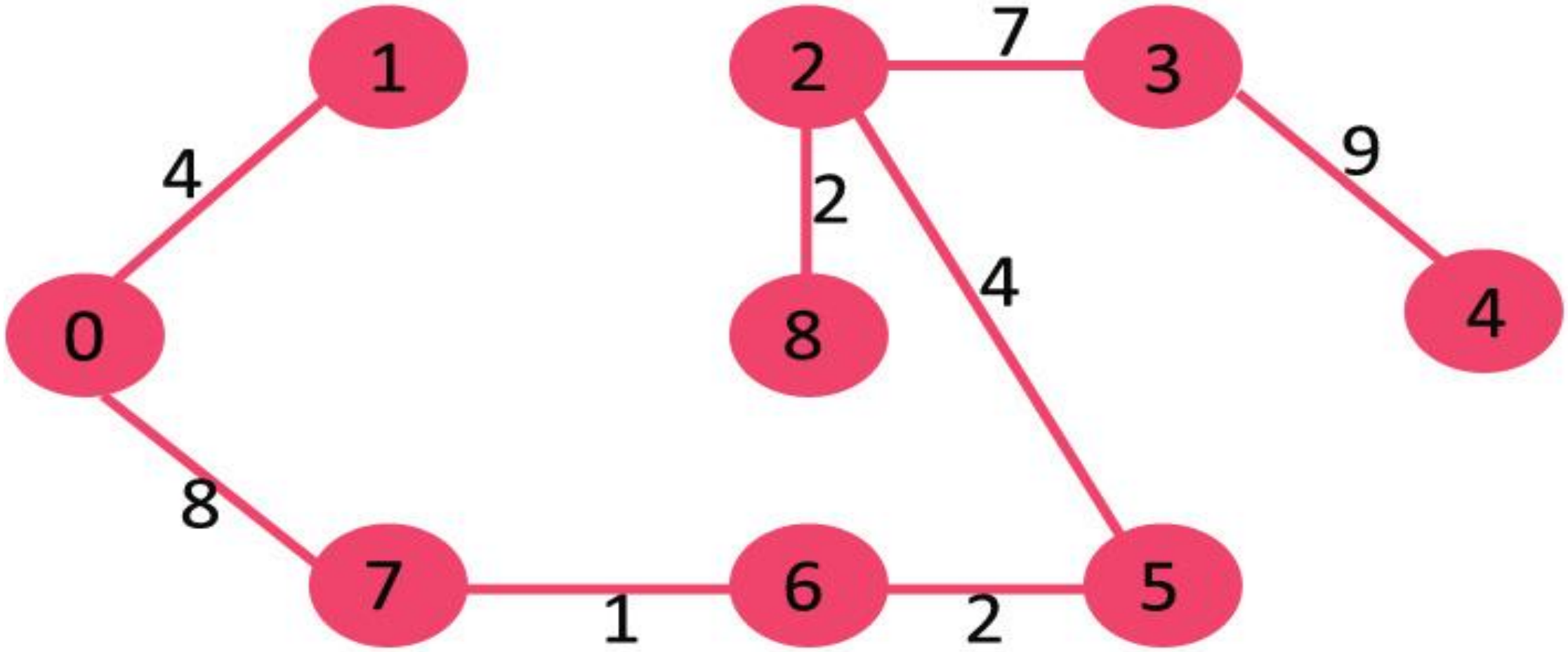
8. *Pick edge 7-8:* Since including this edge results in cycle, discard it.

9. *Pick edge 0-7:* No cycle is formed, include it.



10. *Pick edge 1-2:* Since including this edge results in cycle, discard it.

11. *Pick edge 3-4:* No cycle is formed, include it.



Total Weight of the above spanning tree is 4+8+1+2+4+2+7+9=37

Prim's algorithm

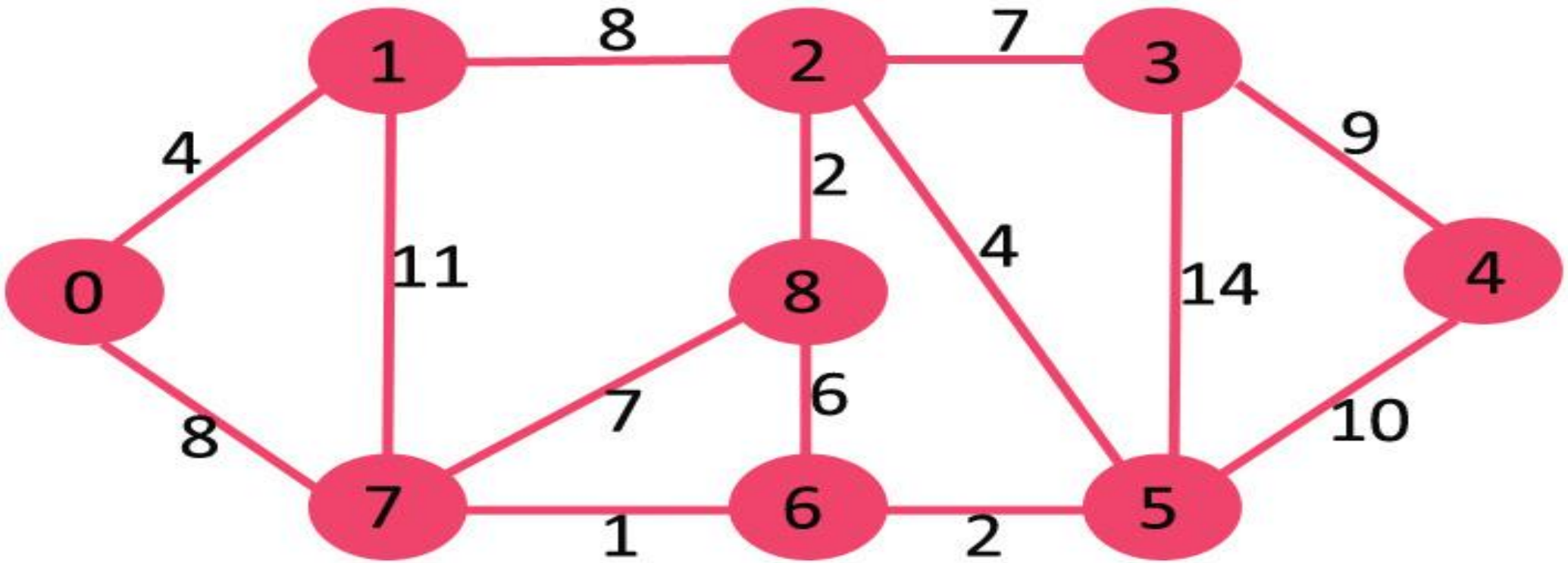
Rather than build a subgraph one edge at a time, Prim's algorithm builds a tree one vertex at a time.

let T be a single vertex x

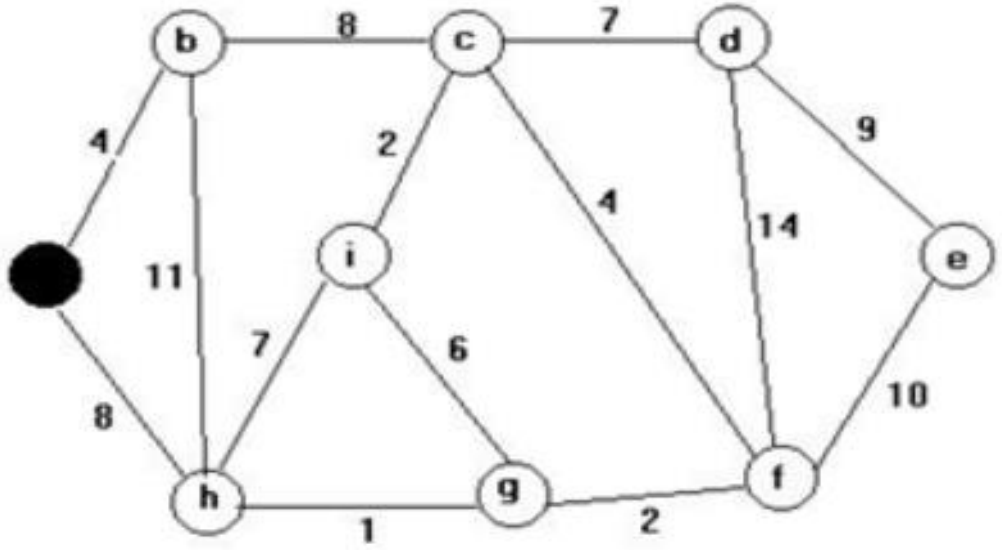
```
while (T has fewer than n vertices)
{
    find the smallest edge connecting T to G-T
    add it to T
}
```

Example: Prim's algorithm

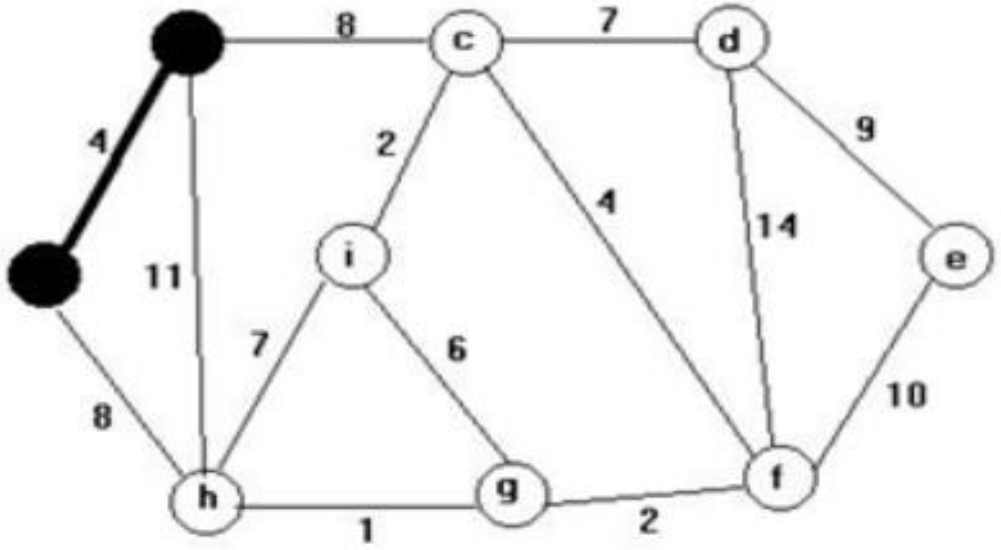
Consider the following connected graph:



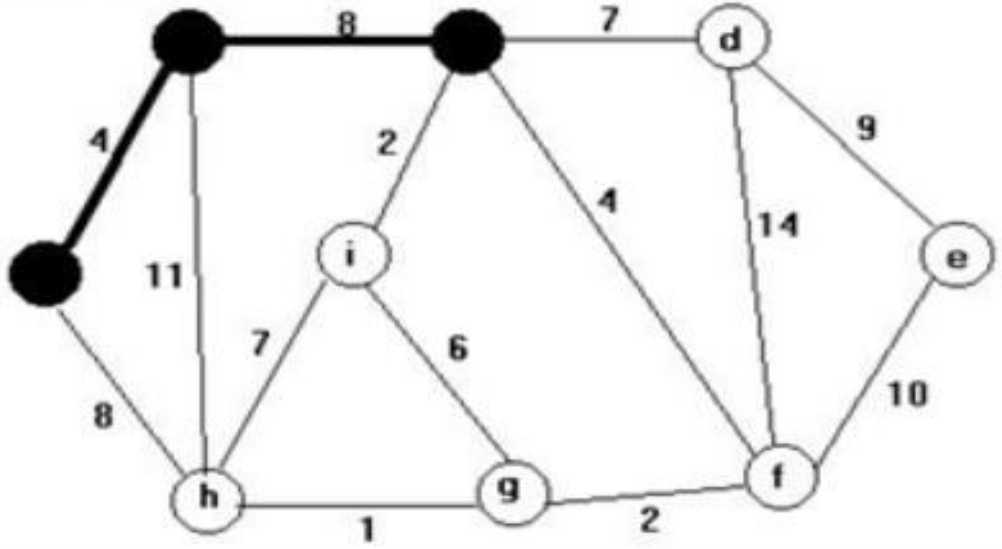
Iteration 0:



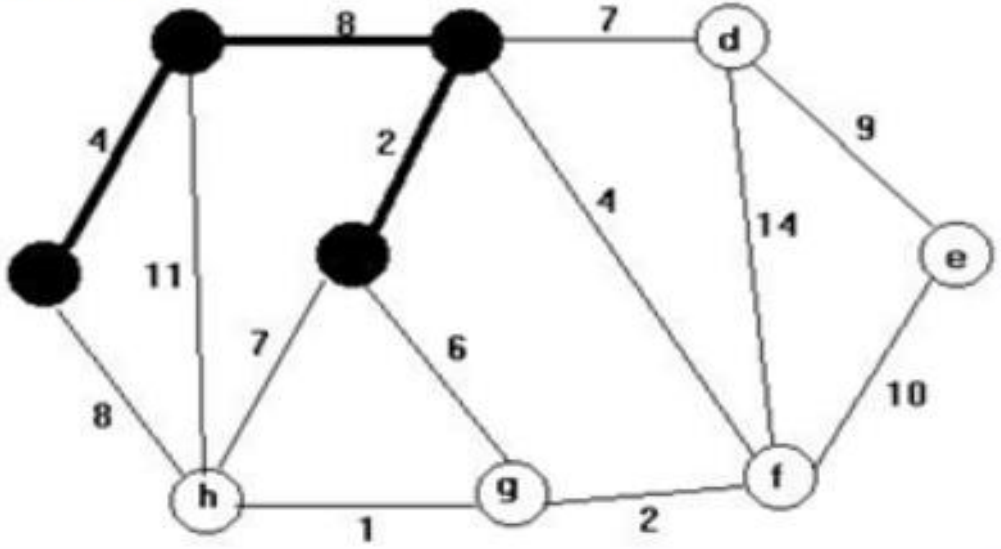
Iteration 1:



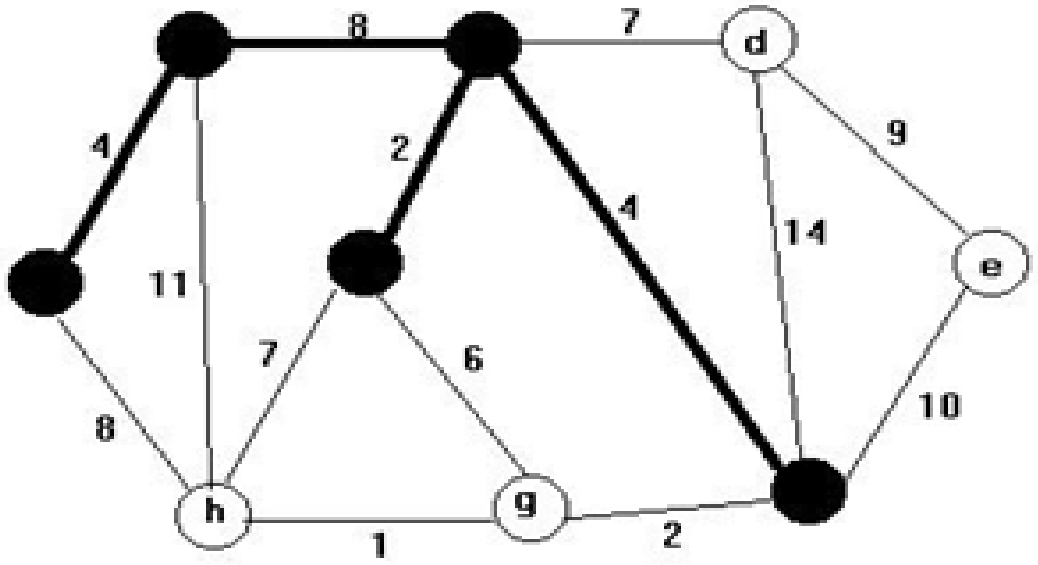
Iteration 2:



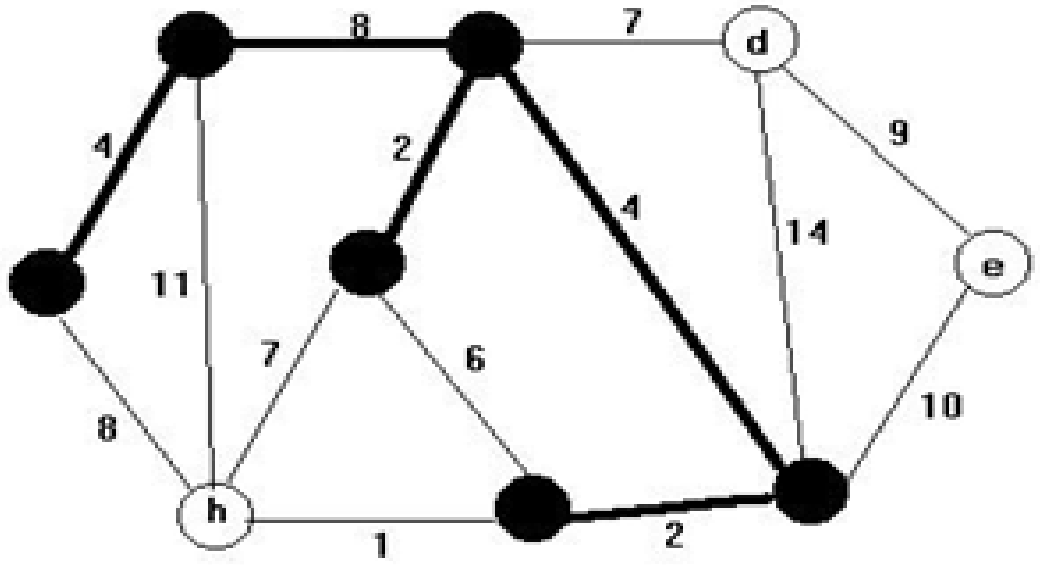
Iteration 3:



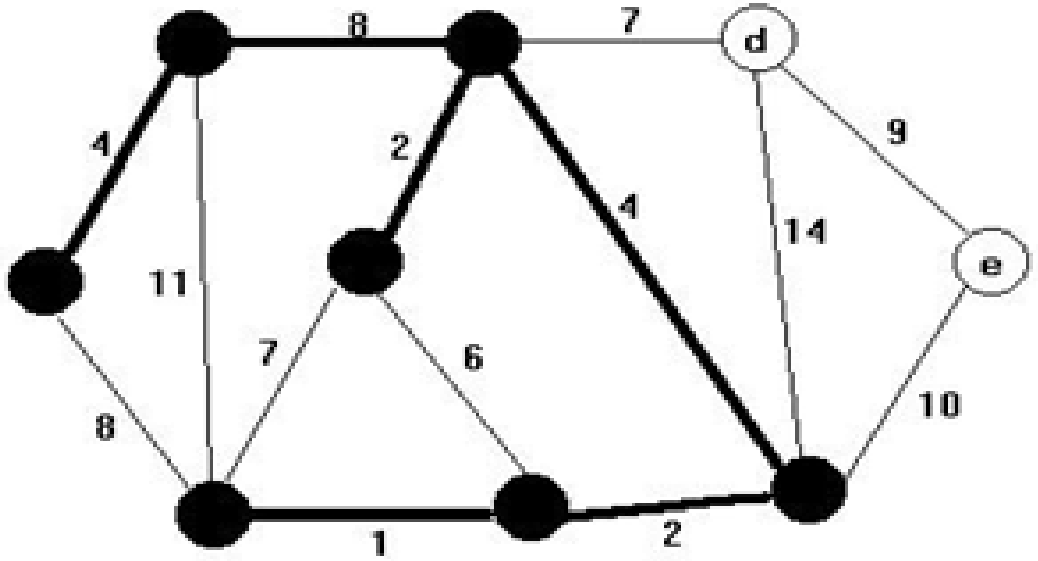
Iteration 4:



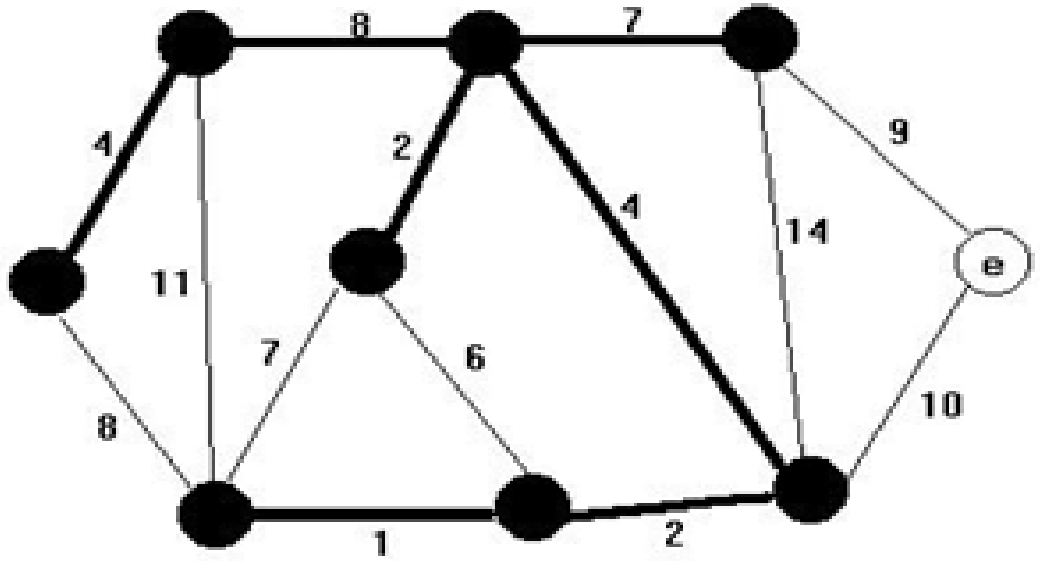
Iteration 5:



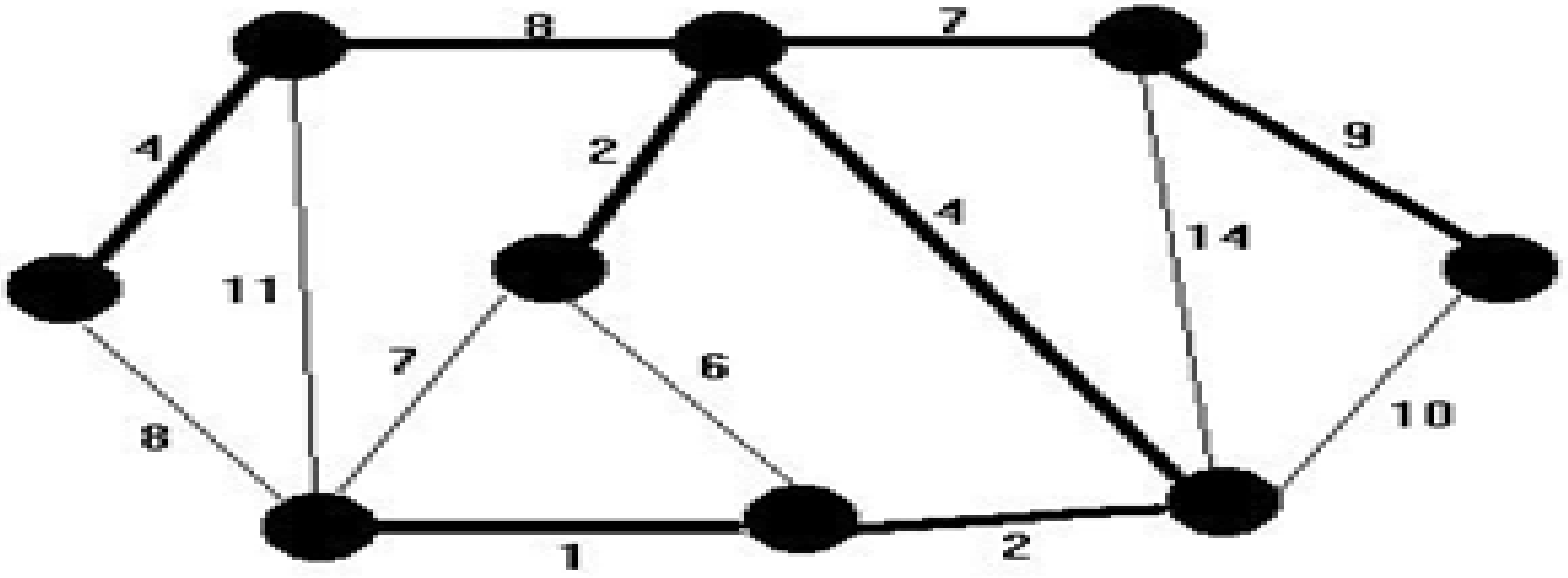
Iteration 6:



Iteration 7:



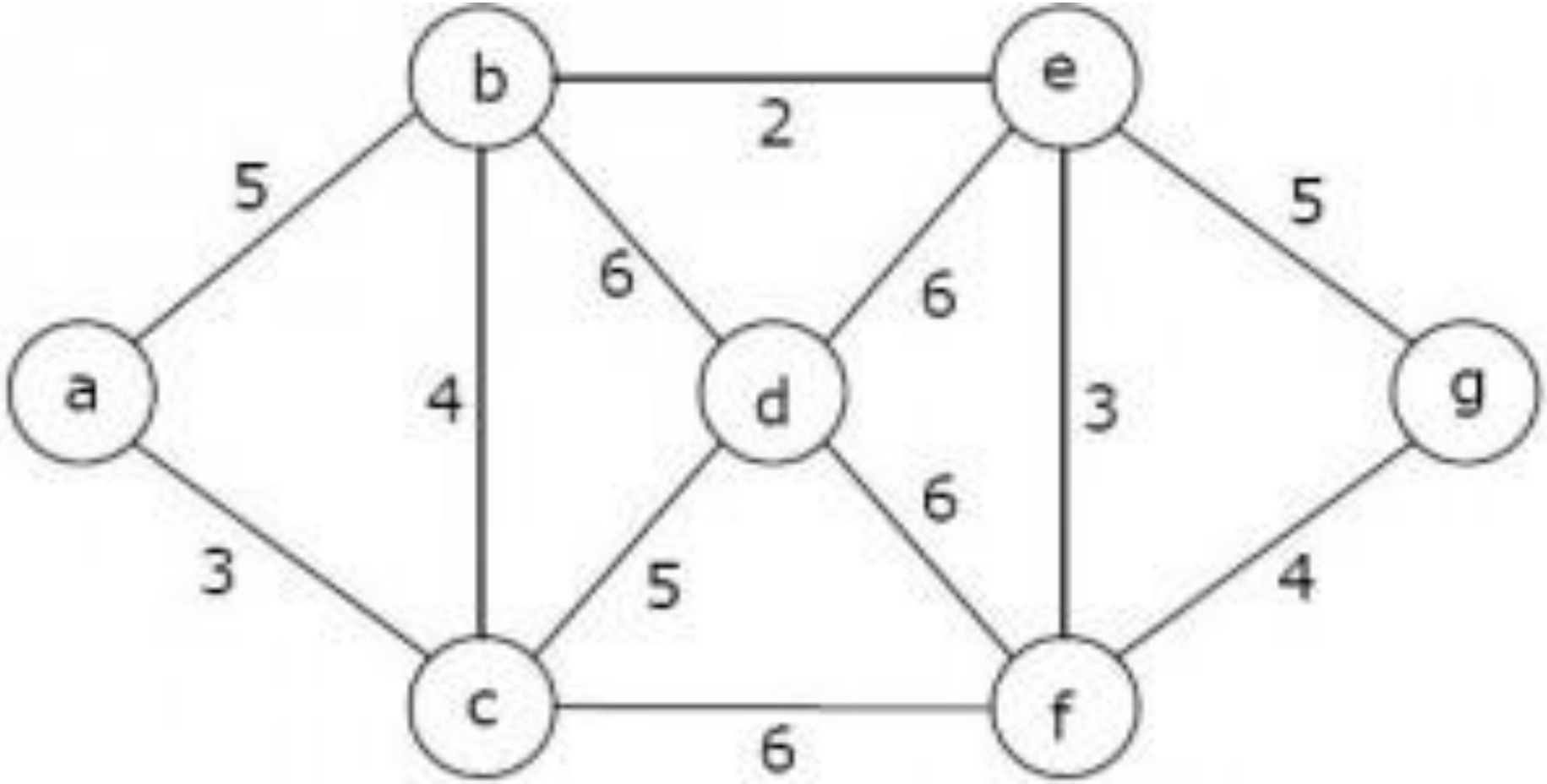
Iteration 8:



Here, the last iteration number 8 represents the optimal solution of the problem, given by the bold dark black lines, which gives us the minimum length of the path. And, the length would be calculated by adding the numbers written on optimal path. That is, 4 + 8 + 7 + 9 + 2 + 4 + 2 + 1 = 37 is the minimum length.

Class Exercise 1

Consider the following graph. Find MST using Kruskal's and Prim's algorithm. Also find the total cost of MST.

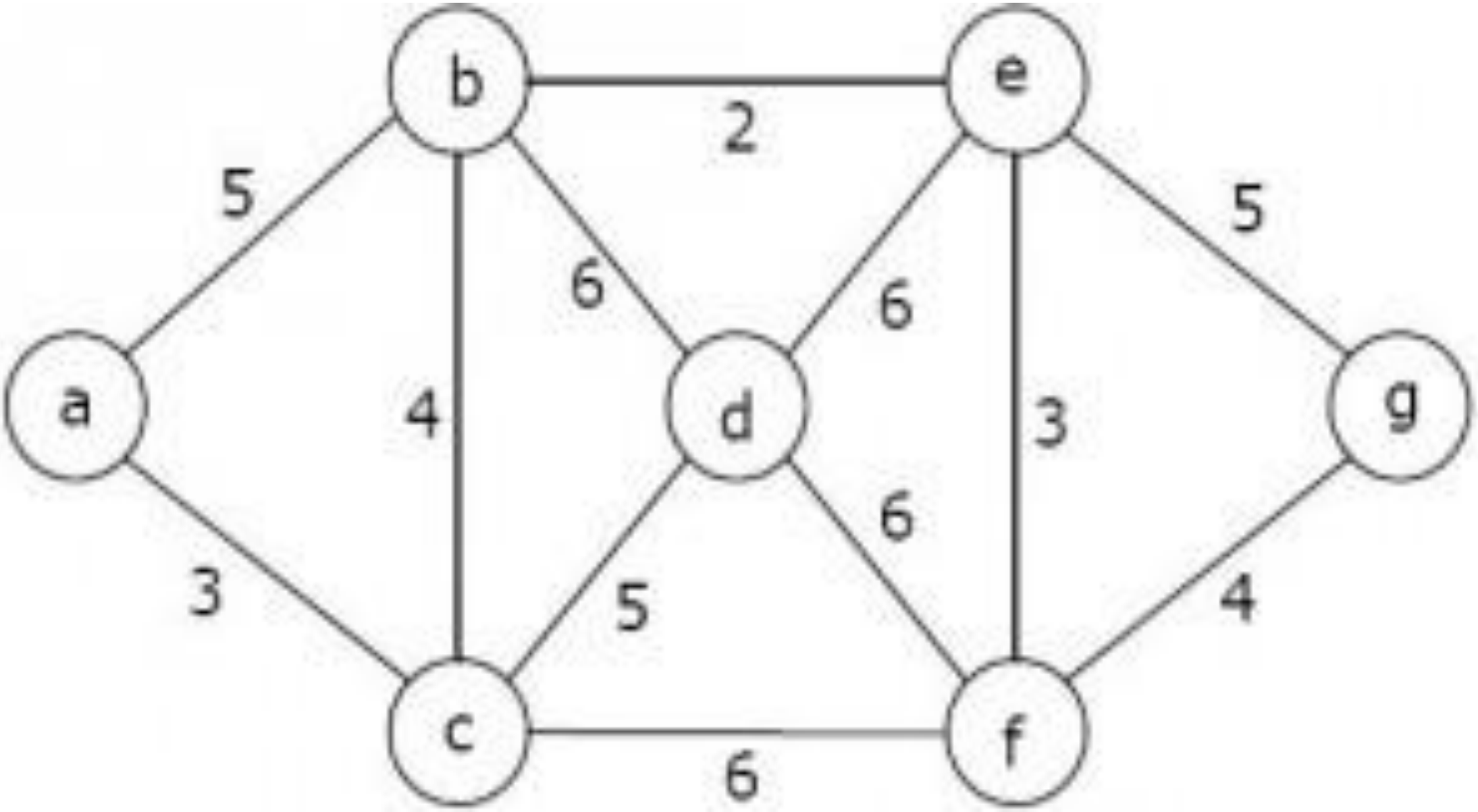


Class Exercise 2

Consider the following graph

Which one of the following is NOT the sequence of edges added to the minimum spanning tree using Kruskal's algorithm?

(A) (b,e)(e,f)(a,c)(b,c)(f,g)(c,d)
(B) (b,e)(e,f)(a,c)(f,g)(b,c)(c,d)
(C) (b,e)(a,c)(e,f)(b,c)(f,g)(c,d)
(D) (b,e)(e,f)(b,c)(a,c)(f,g)(c,d)


b
e
2
5
5
6
6
a
4
d
3
g
6
3
5
4
c
f
6

Class Exercise 3

Consider a complete undirected graph with vertex set {0, 1, 2, 3, 4}. Entry Wij in the matrix W below is the weight of the edge {i, j}.

What is the minimum possible weight of a spanning tree T in this graph such that vertex 0 is a leaf node in the tree T?

$$W = \begin{pmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{pmatrix}$$

Class Exercise 4

use Prim's algorithm to find a minimum spanning tree for the given weighted graph.