# Design and Analysis of Algorithms

# Lecture-28

Dharmendra Kumar (Associate Professor)
Department of Computer Science and Engineering
United College of Engineering and Research,
Prayagraj

# Minimum Spanning Tree

## Spanning tree:

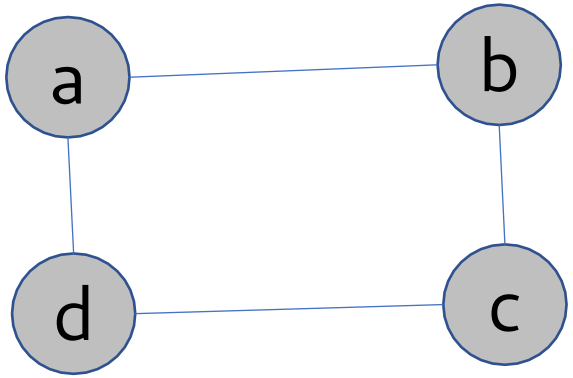
Spanning tree is a non-cyclic sub-graph of a connected and undirected graph G that connects all the vertices together.

## General Properties of Spanning Tree

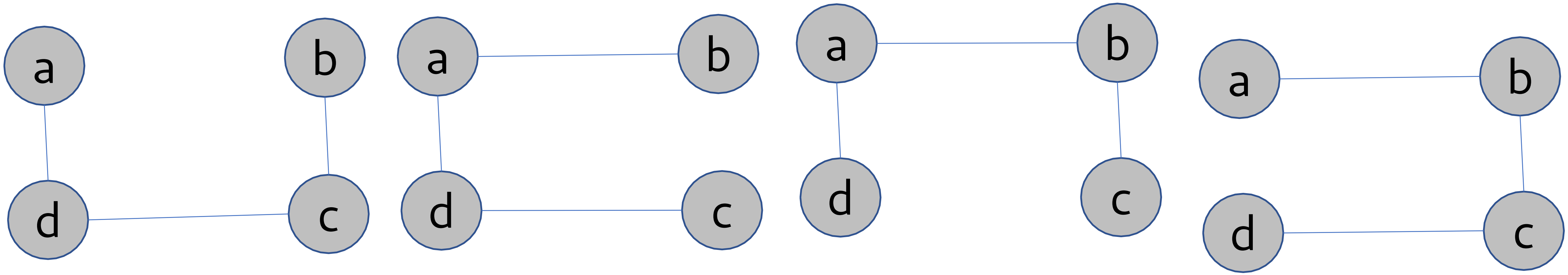
- A connected graph G can have more than one spanning tree.
- All possible spanning trees of graph G, have the same number of edges and vertices.
- The spanning tree does not have any cycle (loops).
- Removing one edge from the spanning tree will make the graph disconnected, i.e. the spanning tree is **minimally connected**.
- Adding one edge to the spanning tree will create a circuit or loop.
- Spanning tree has **n-1** edges, where **n** is the number of nodes (vertices).
- A complete graph can have maximum $\mathbf{n^{n-2}}$ number of spanning trees.

# Minimum Spanning Tree

**Example:** Consider the following graph:-




All the spanning tree of this graph are the following:-

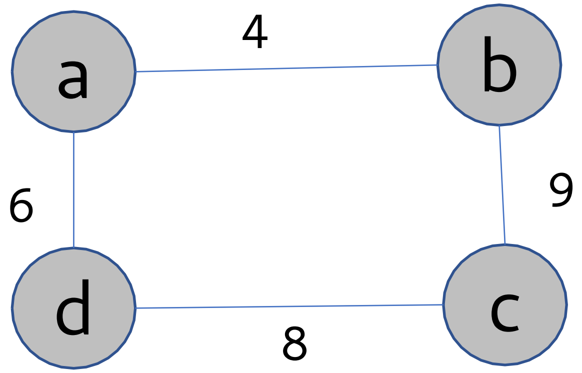


# Minimum Spanning Tree

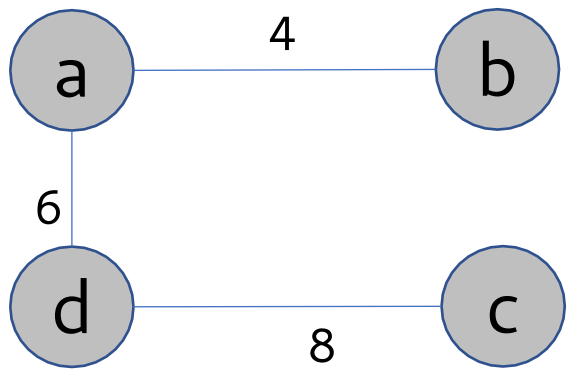
If the given graph is weighted graph, then we define the minimum spanning tree.

**Definition:** A spanning tree is said to be minimum spanning tree if sum of weights of all the edges in the tree is smallest.

**Example:** Consider the following graph:–




Graph




Total cost = 18

Minimum spanning tree

# Minimum Spanning Tree

- In this chapter, we will study two algorithms to find the minimum spanning tree.

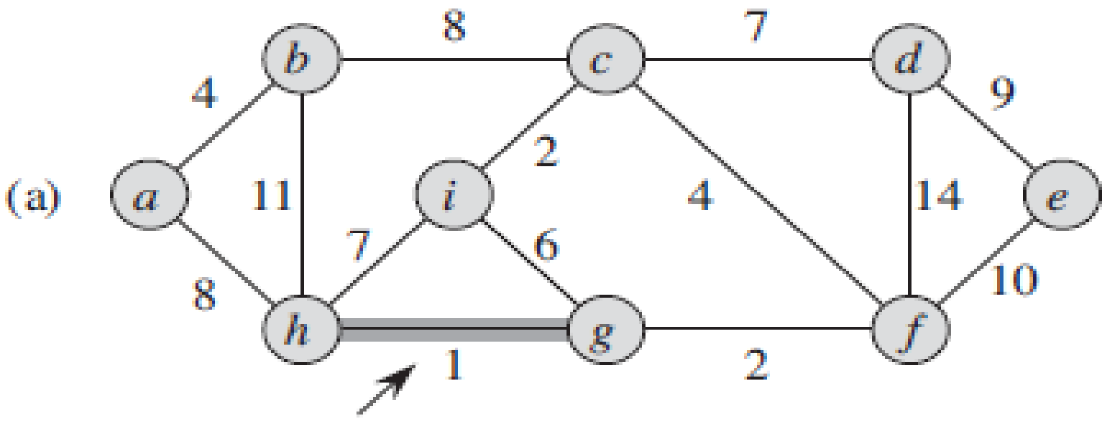
(1) Kruskal's Algorithm

(2) Prim's Algorithm

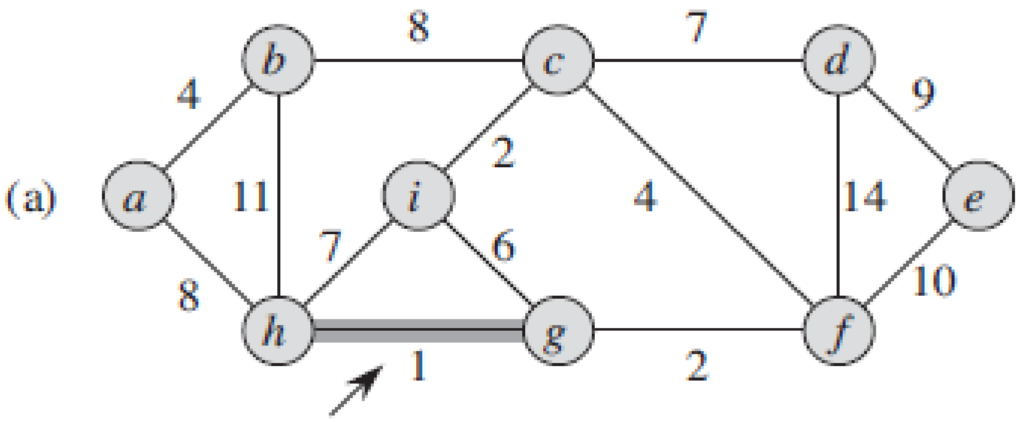
Both algorithms are based on Greedy approach.

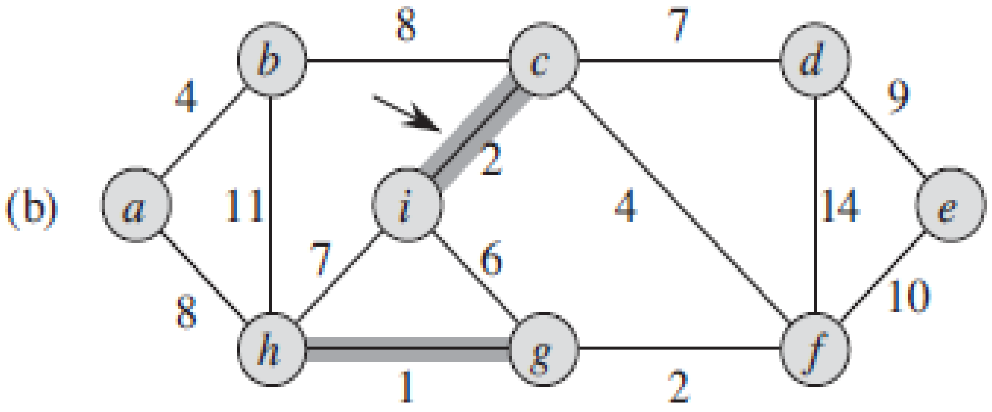
# Kruskal’s Algorithm

Example: Find the minimum spanning of the following graph using Kruskal algorithm.

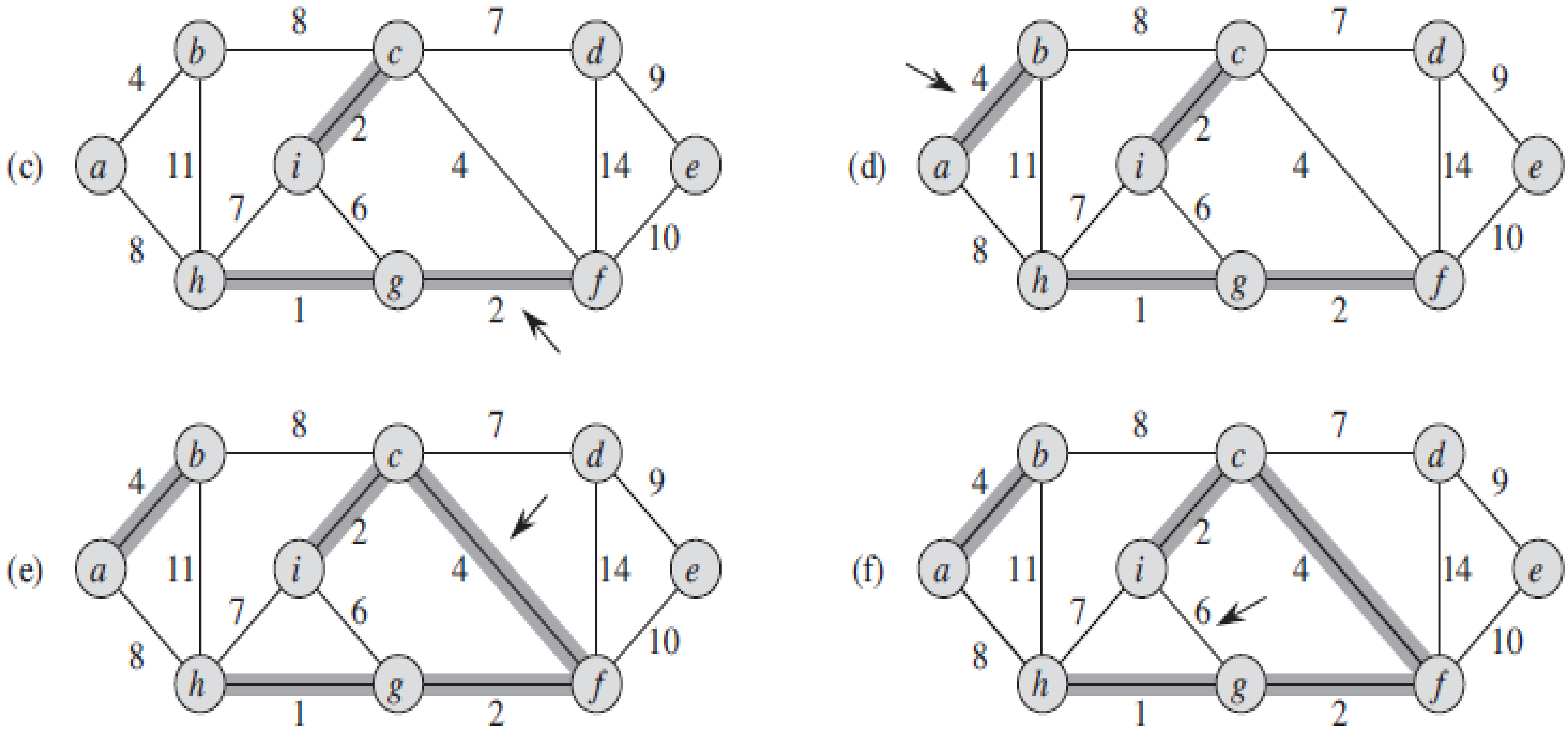



Solution:

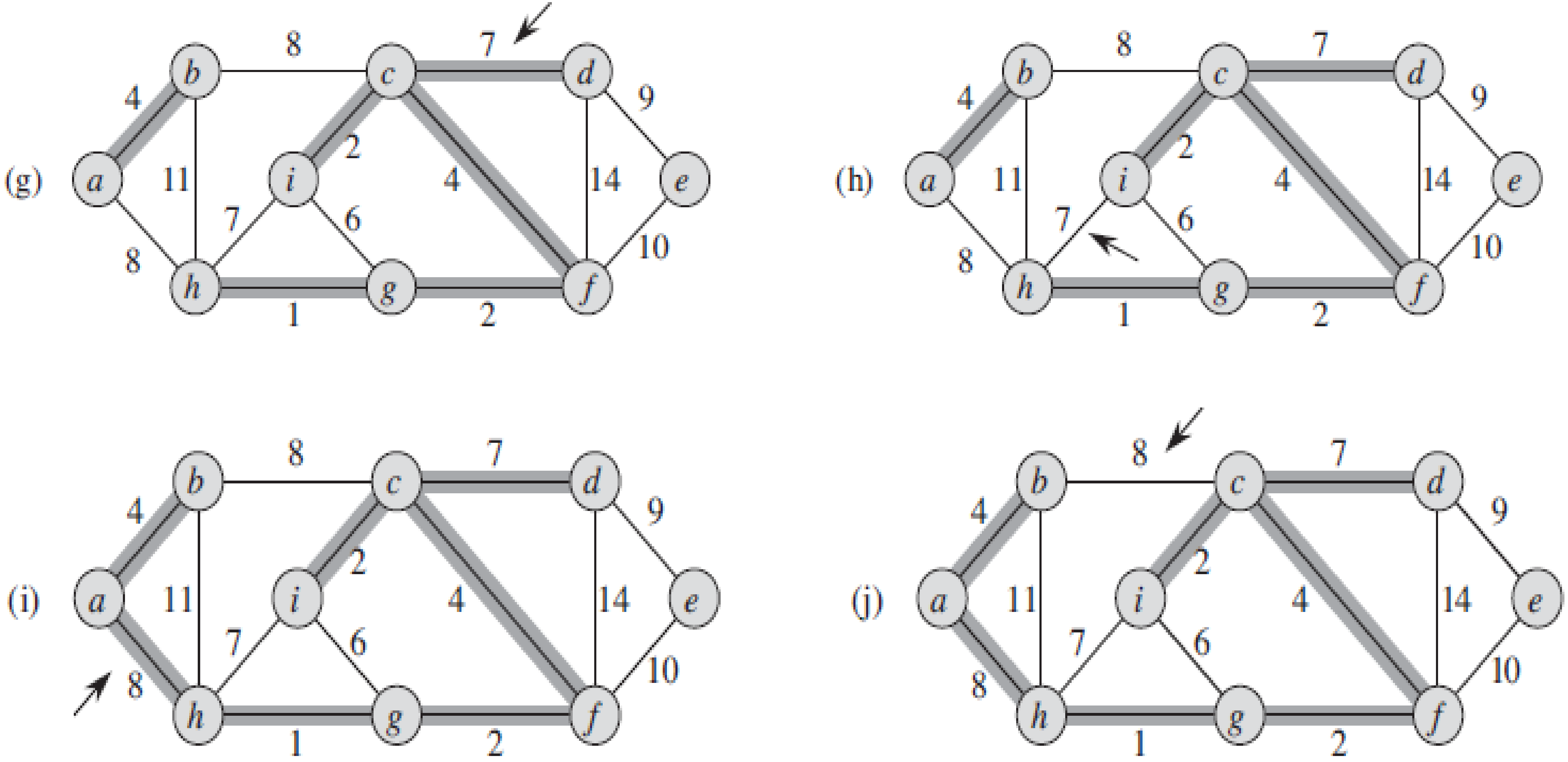


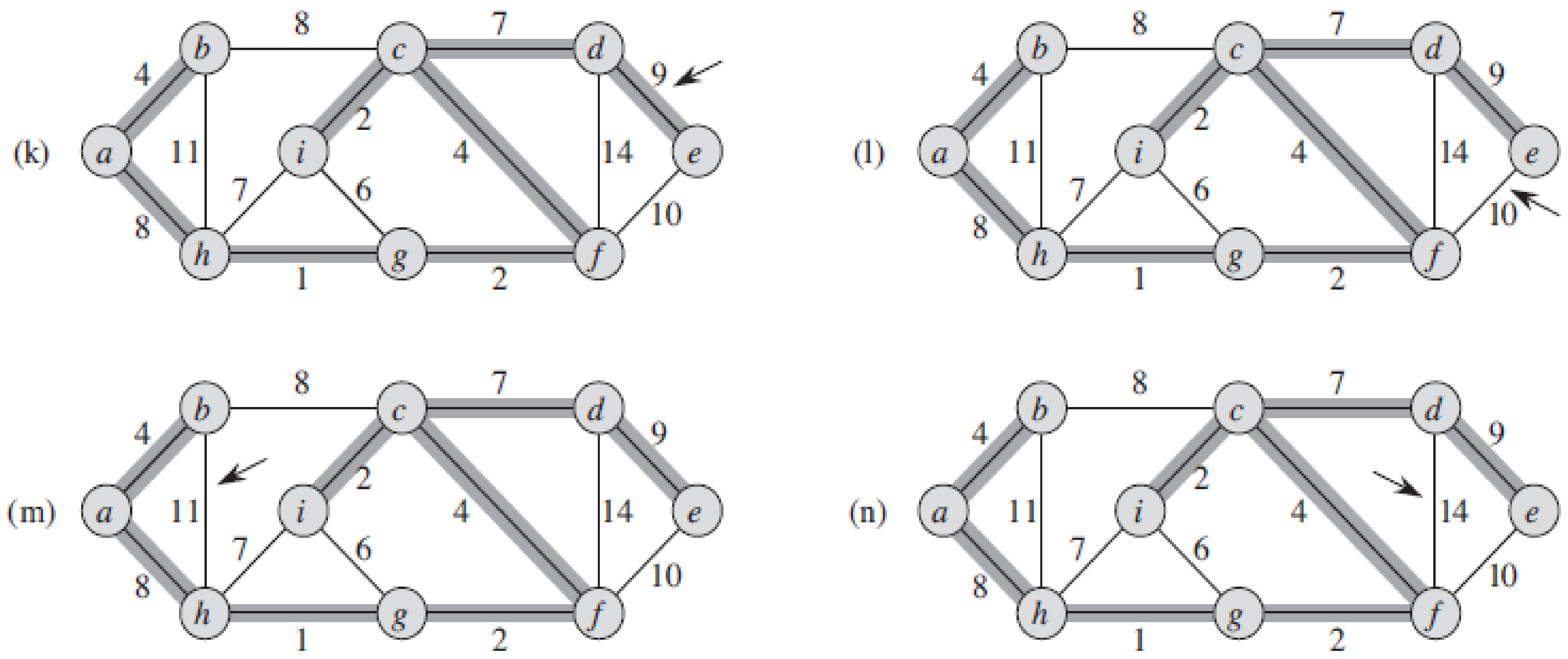
# Kruskal's Algorithm



# Kruskal’s Algorithm



# Kruskal’s Algorithm




Total cost = 37

Final minimum spanning tree

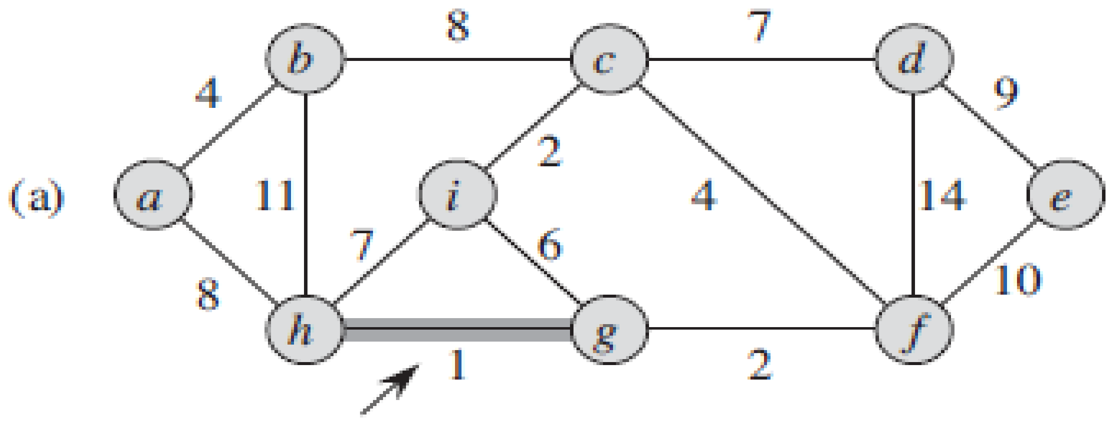
# Kruskal's Algorithm

```
MST-KRUSKAL(G, w)
A = ∅
for each vertex v ∈ G.V
    MAKE-SET(v)
sort the edges of G.E into nondecreasing order by weight w
for each edge (u, v) ∈ G.E, taken in nondecreasing order by weight
    if FIND-SET(u) ≠ FIND-SET(v)
        A = A ∪ {(u, v)}
        UNION(u, v)
return A
```

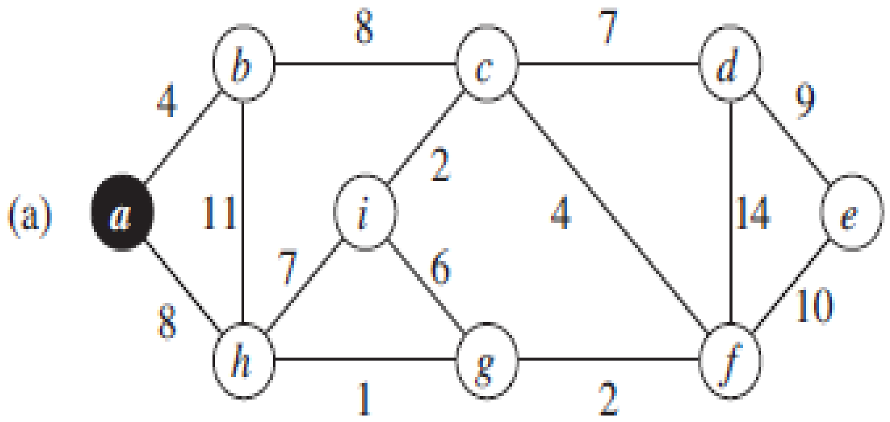
Time complexity of this algorithm is O(E lg E) or O(E lgV).

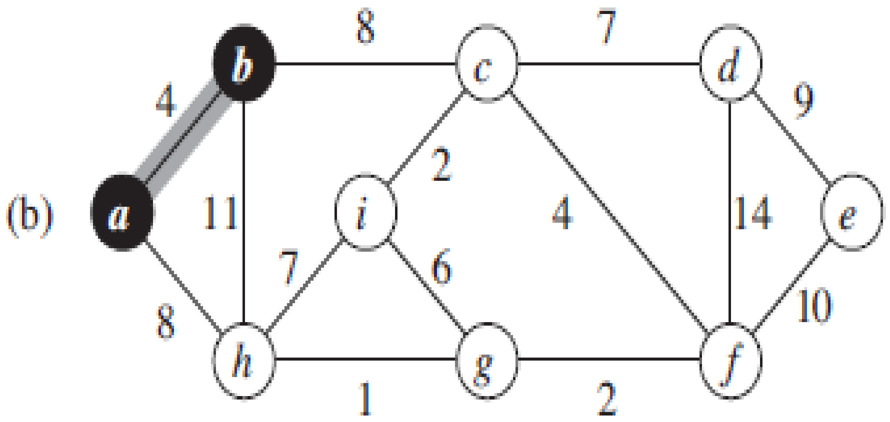
# Prim's Algorithm

Example: Find the minimum spanning of the following graph using Prim's algorithm.

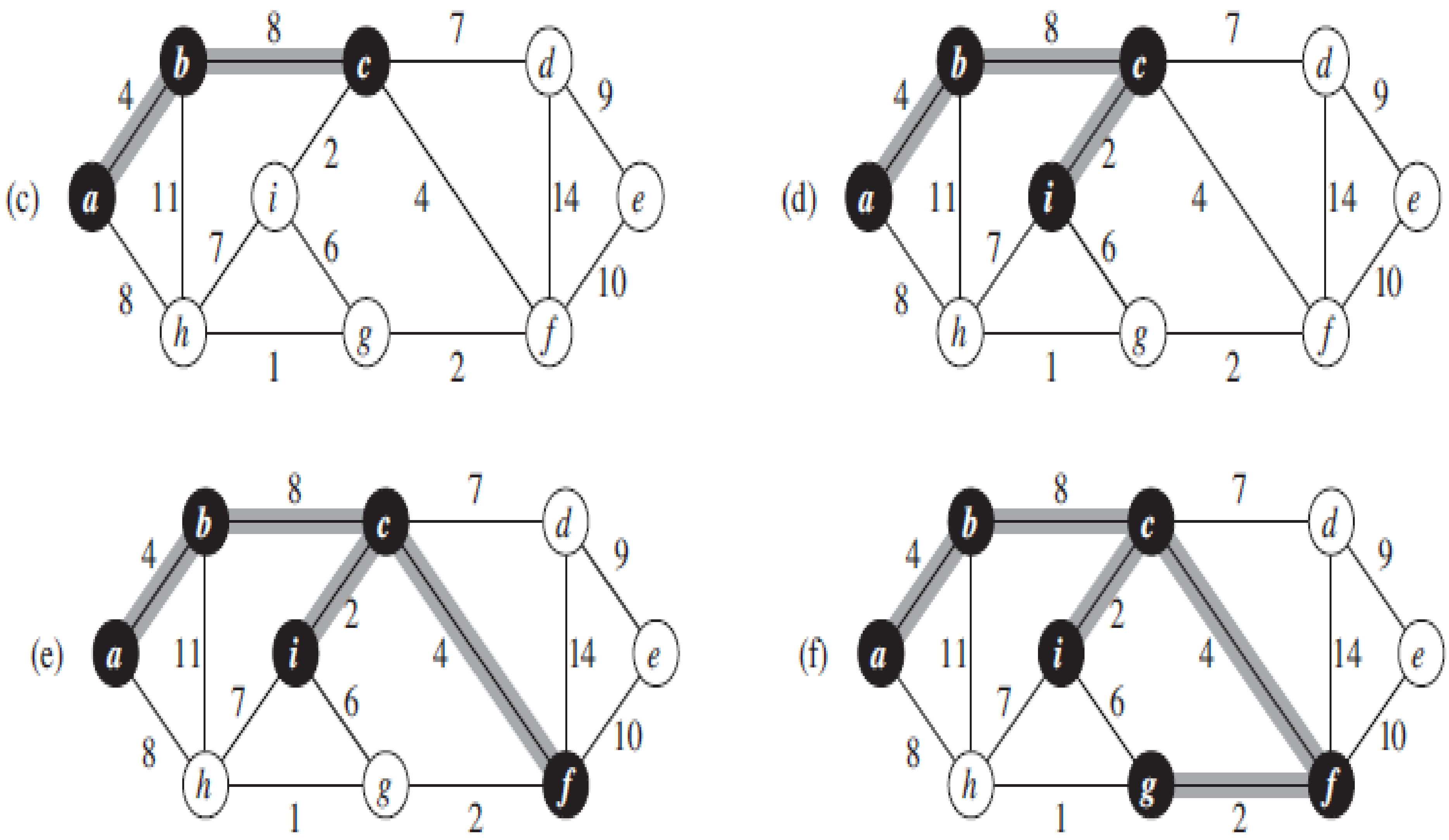



Solution:

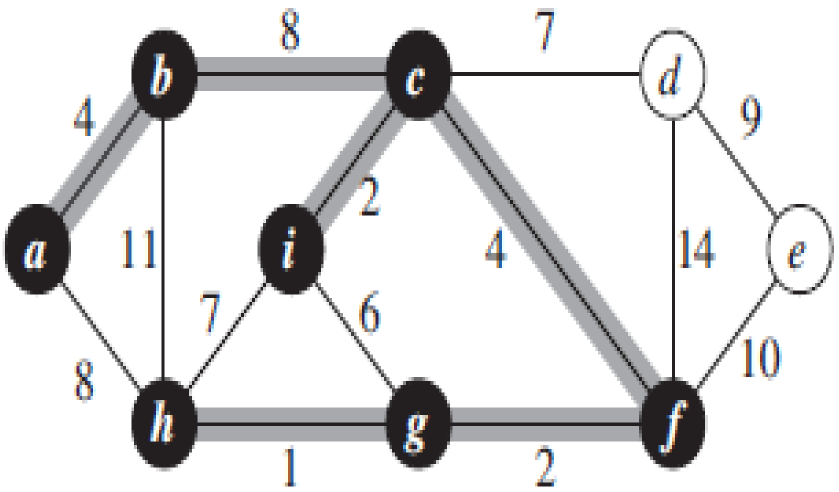


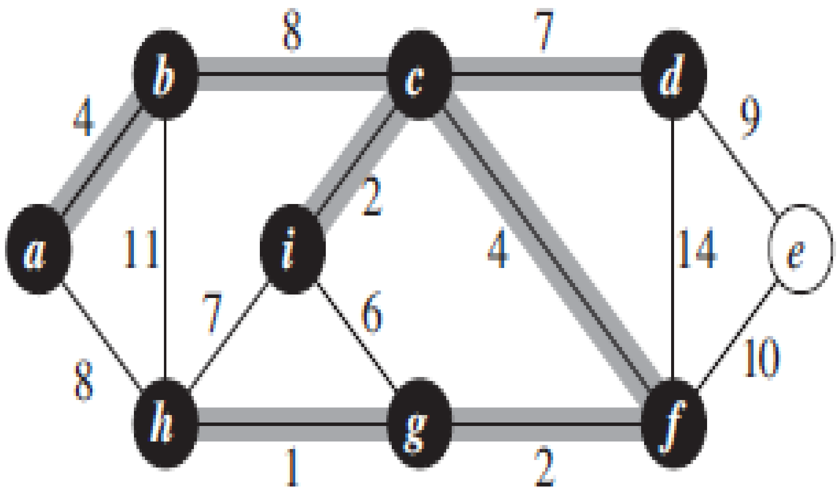



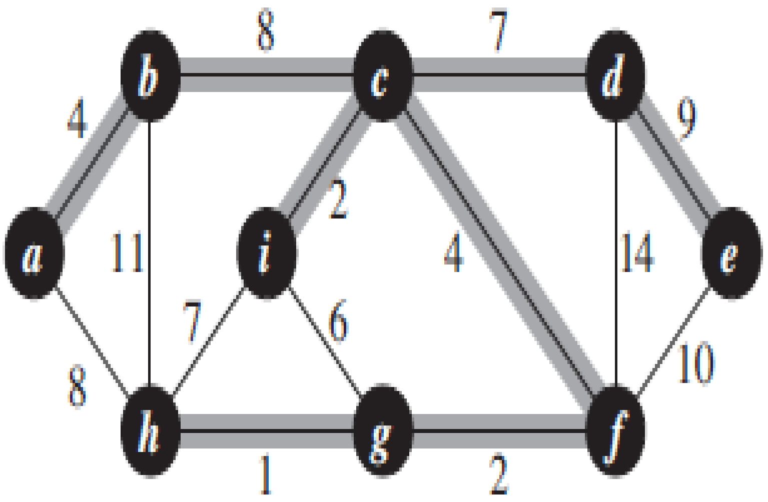
# Prim's Algorithm



# Prim's Algorithm

(g)



(h)



(i)



Total cost = 37

Final minimum spanning tree

# Prim's Algorithm

MST-PRIM$(G, w, r)$

```
for each u ∈ G.V
    u.key = ∞
    u.π = NIL
r.key = 0
Q = G.V
while Q ≠ ∅
    u = EXTRACT-MIN(Q)
    for each v ∈ G.Adj[u]
        if v ∈ Q and w(u, v) < v.key
            v.π = u
            v.key = w(u, v)
```

Time complexity of this algorithm is $O(E \lg V + V \lg V) = O(E \lg V)$.