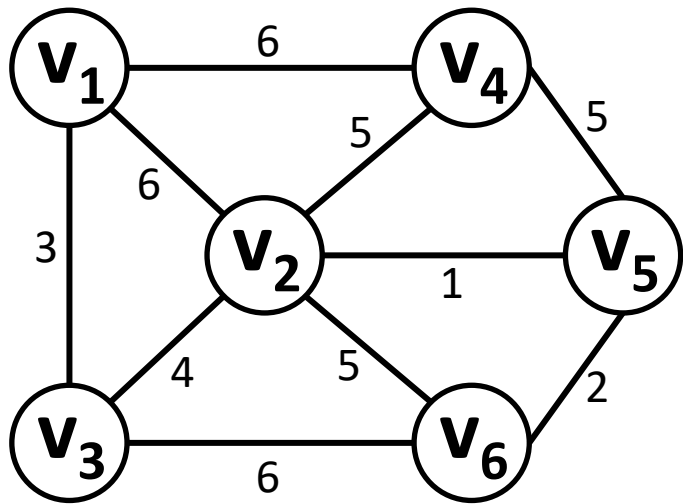
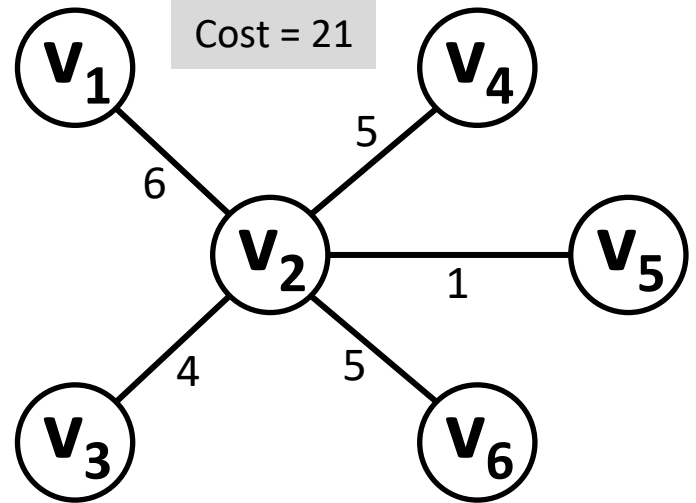


Graphs

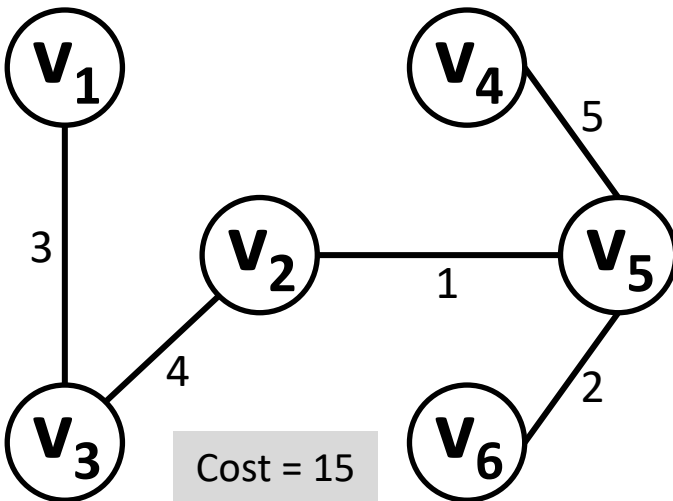
Minimum Spanning Tree



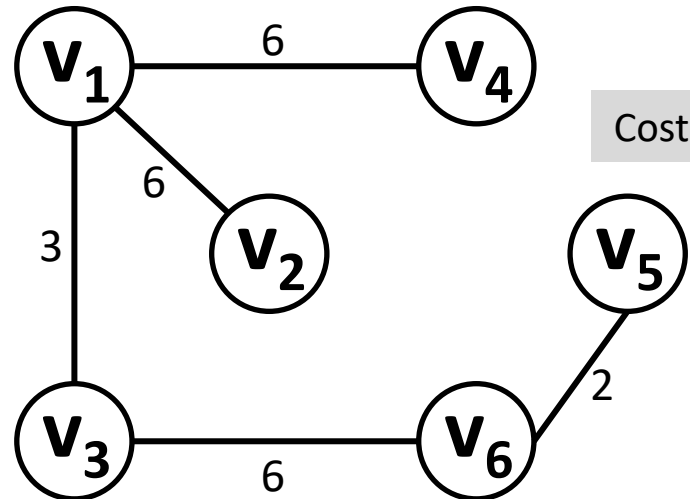
Connected Graph



Cost = 21



Cost = 15



Cost = 23

Introduction

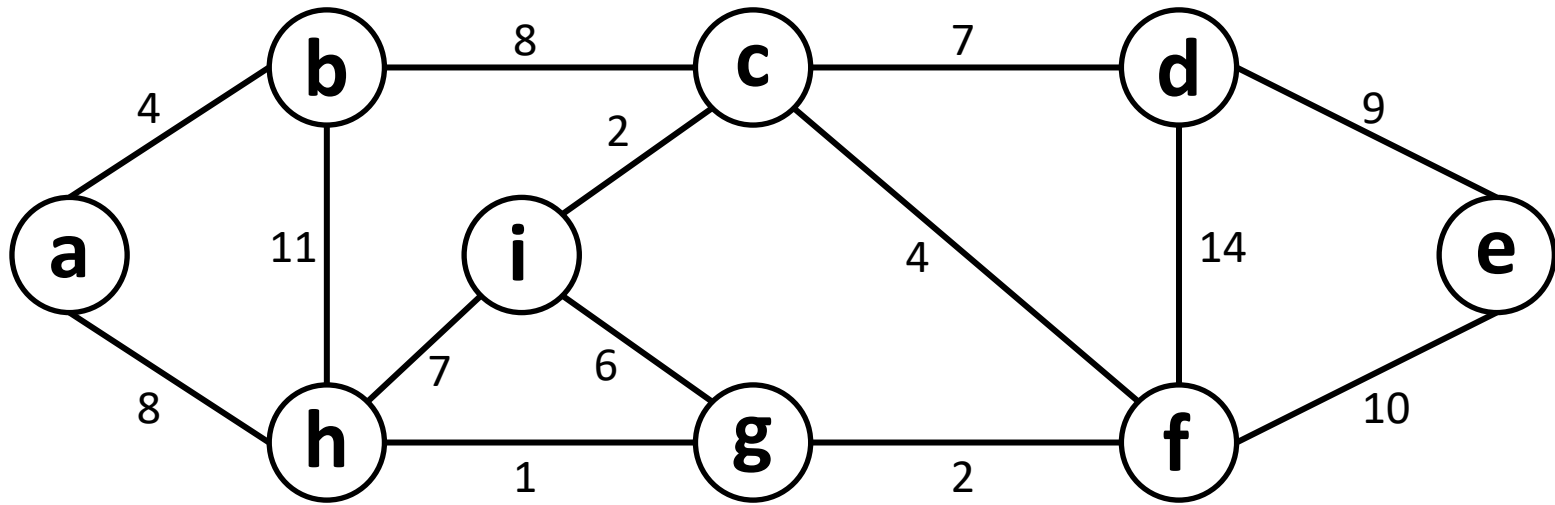
- Let, $G = (V, E)$ be a connected, undirected graph and $w(u, v)$ be a weight/cost of an edge $(u, v) \in E$.
- Then an acyclic subset $T \subseteq E$, that connects all of the vertices in V , is called a "spanning tree".
- The problem termed as "minimum spanning tree" or "minimum-weight spanning tree" aims to minimize total weight given as

$$w(T) = \sum_{(u,v) \in T} w(u, v)$$

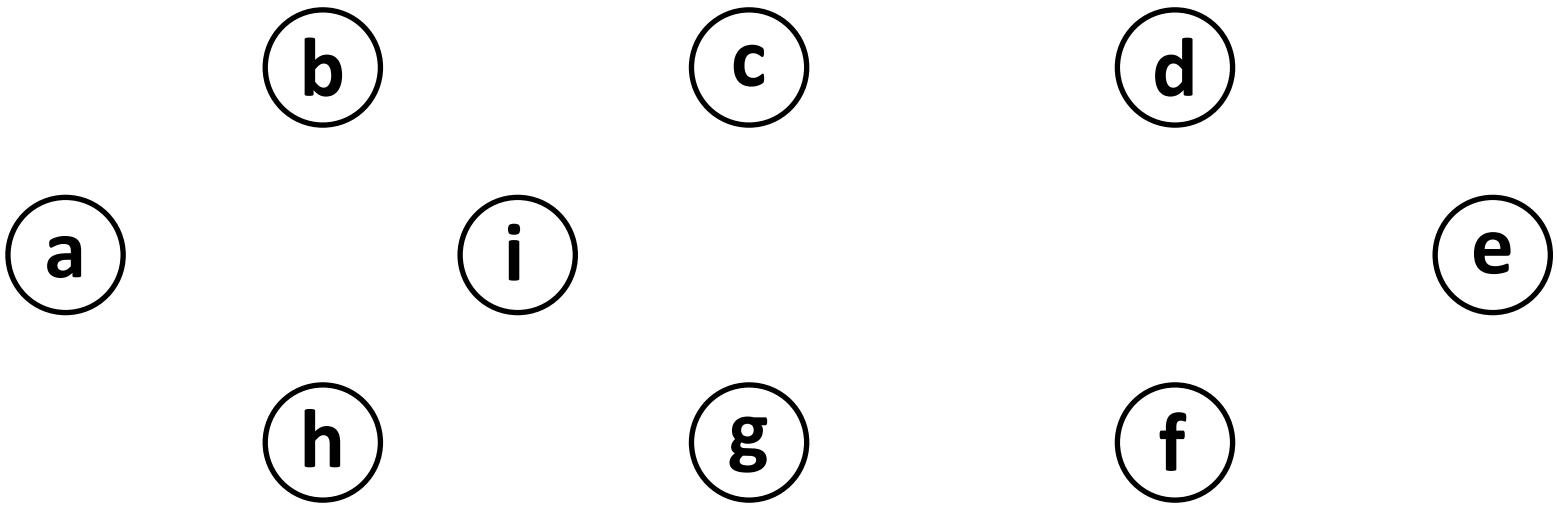
Contd...

- Algorithms to solve the minimum spanning tree problem:
 - Kruskal's algorithm and
 - Prim's algorithm.

Kruskal's Algorithm



Contd...

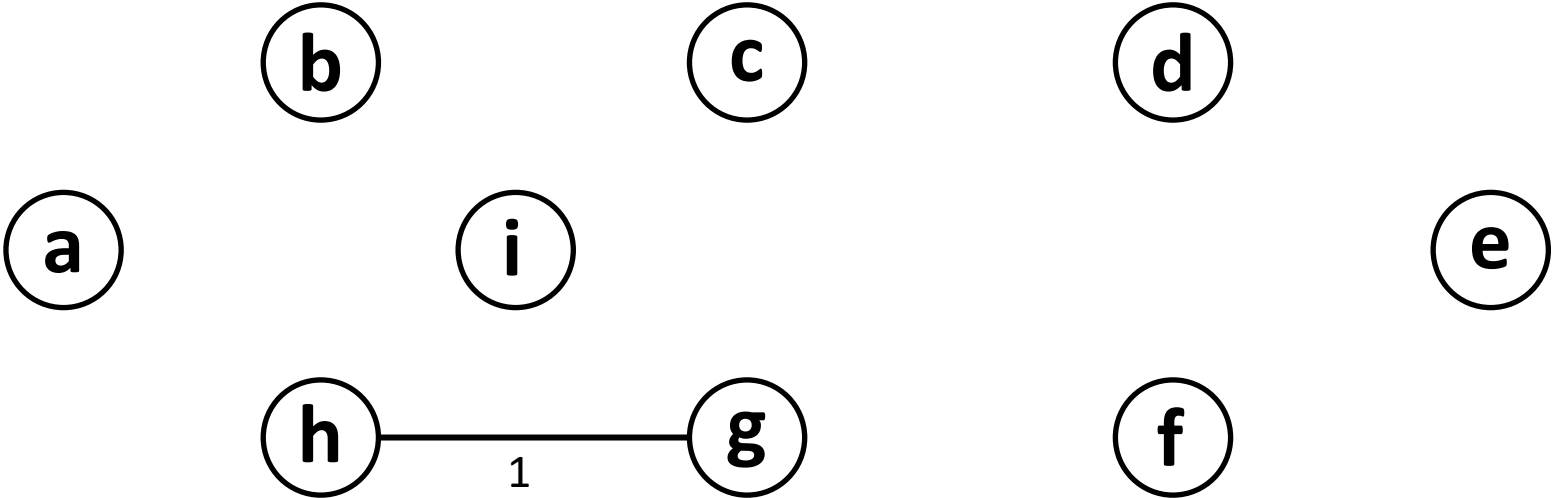


Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

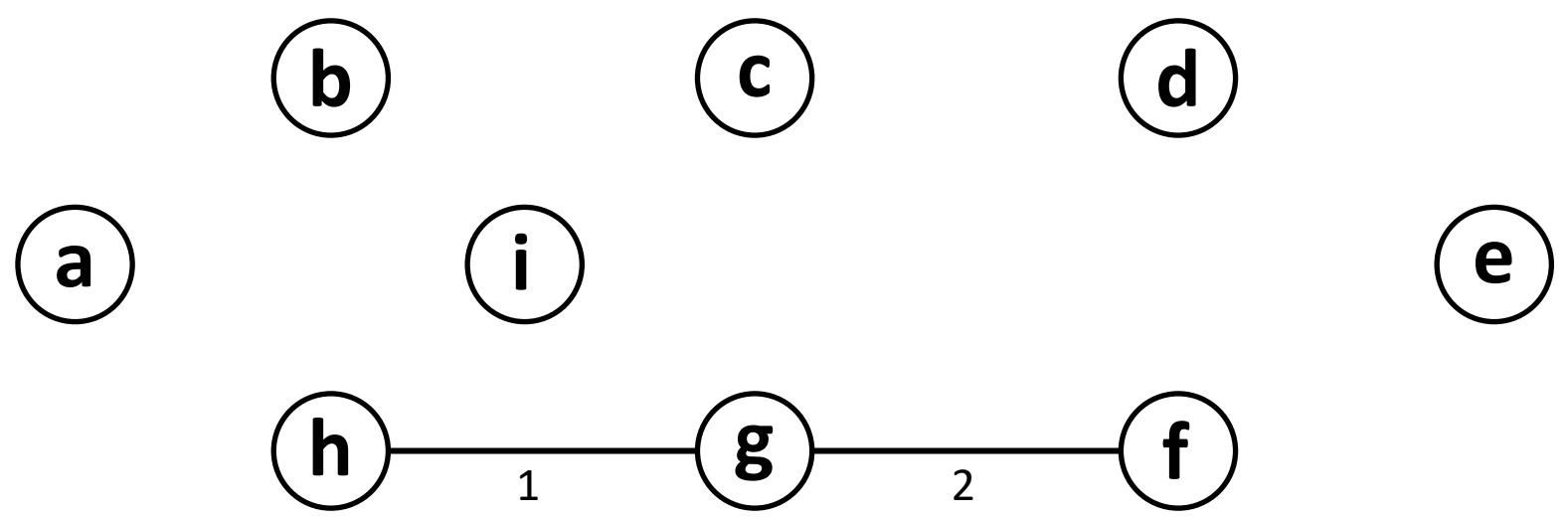


Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

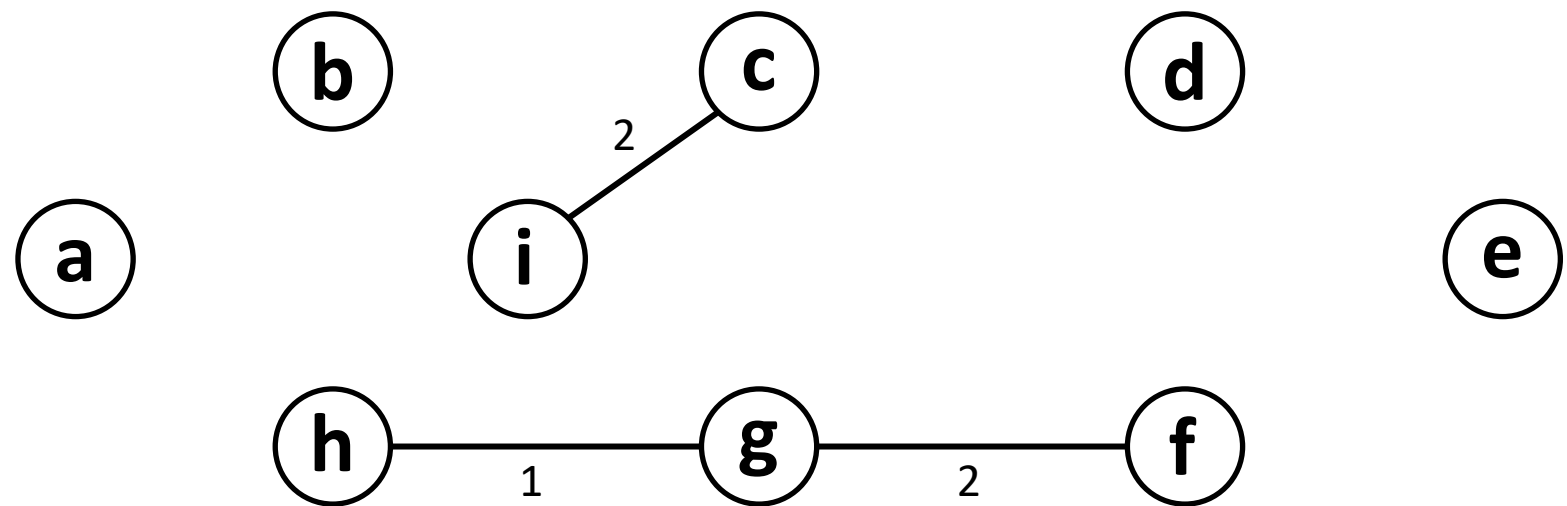


Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

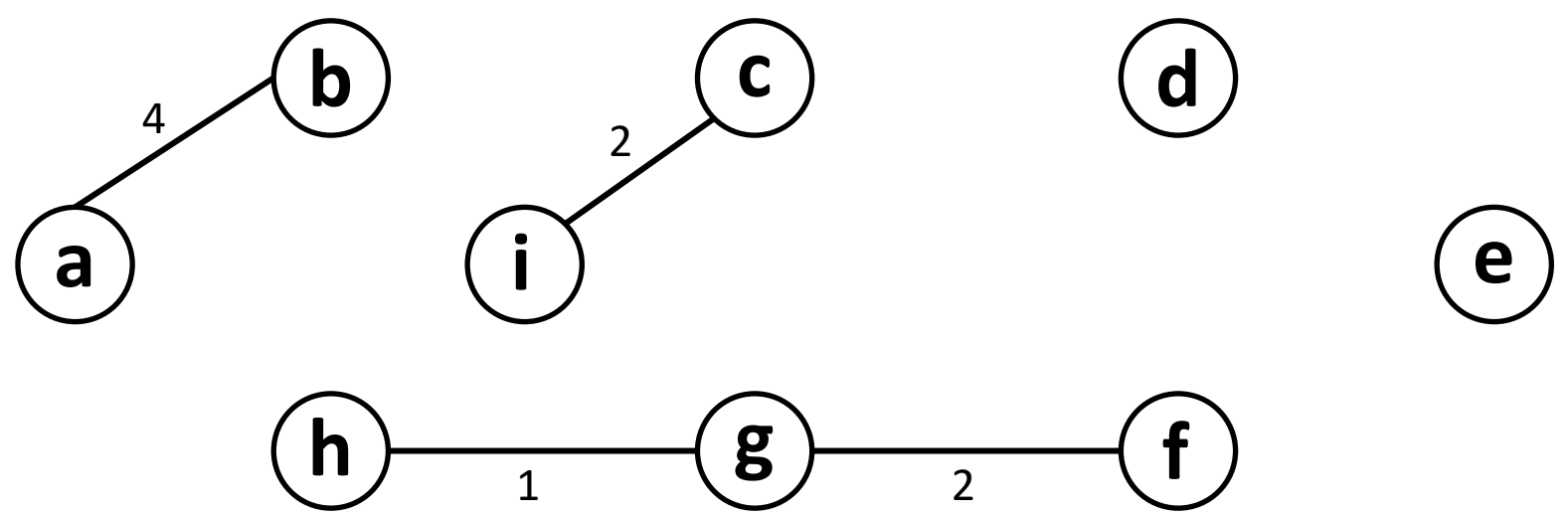


Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

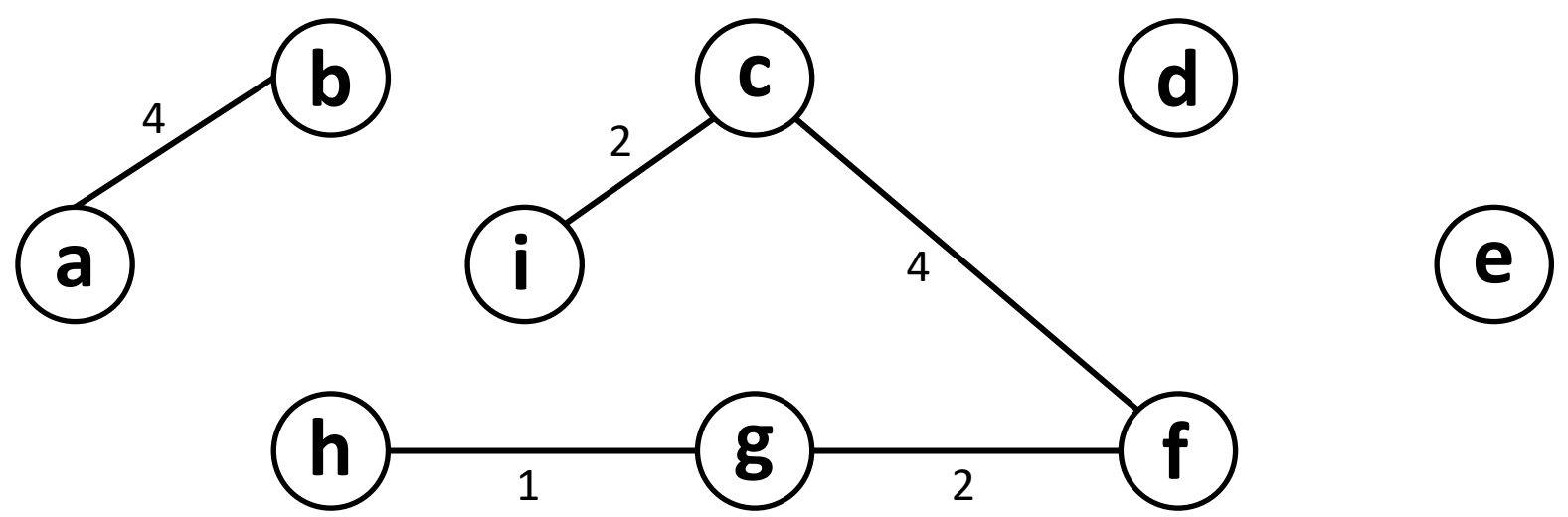


Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

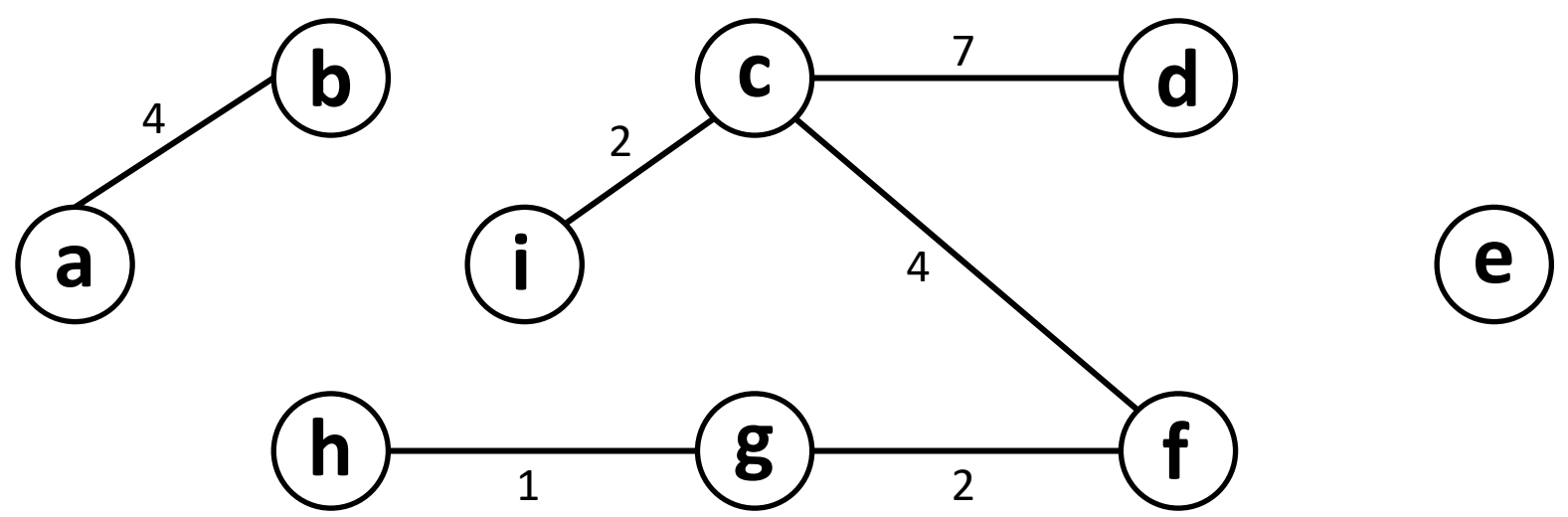


Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

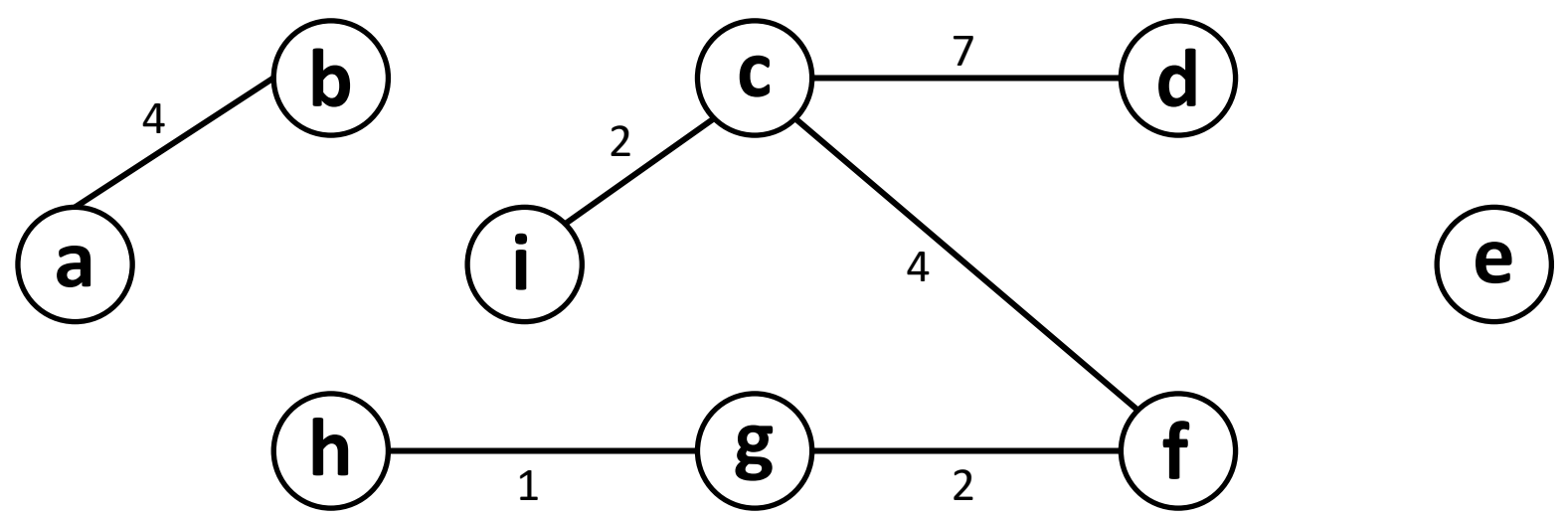


Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

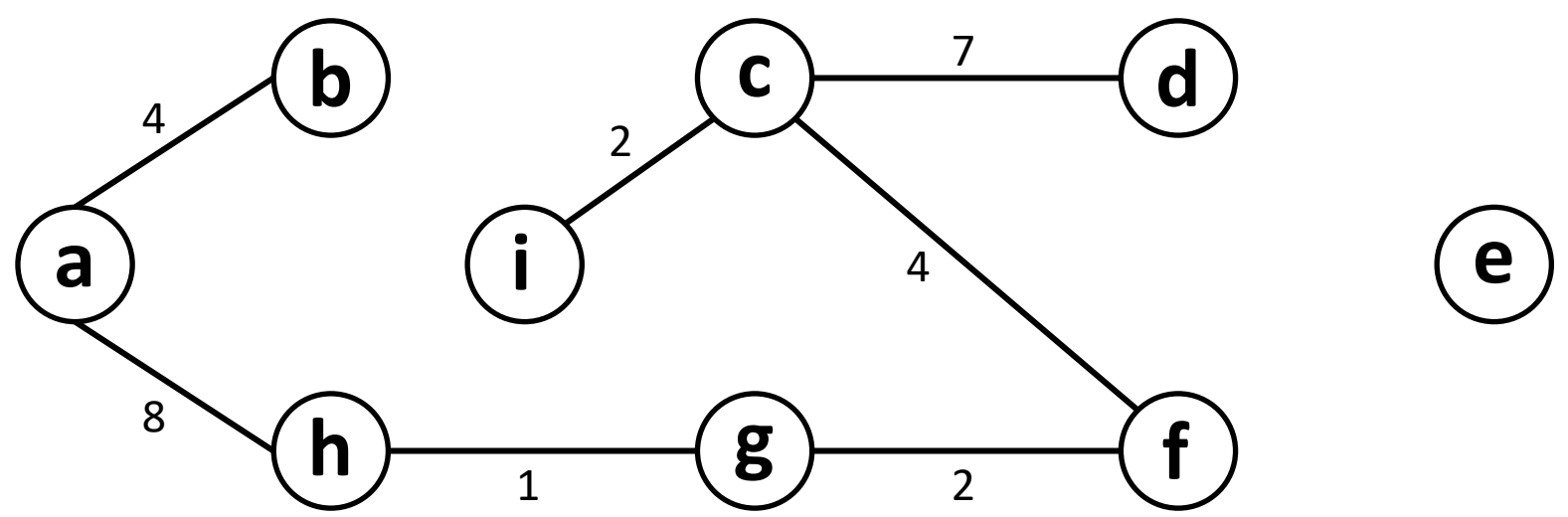


Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

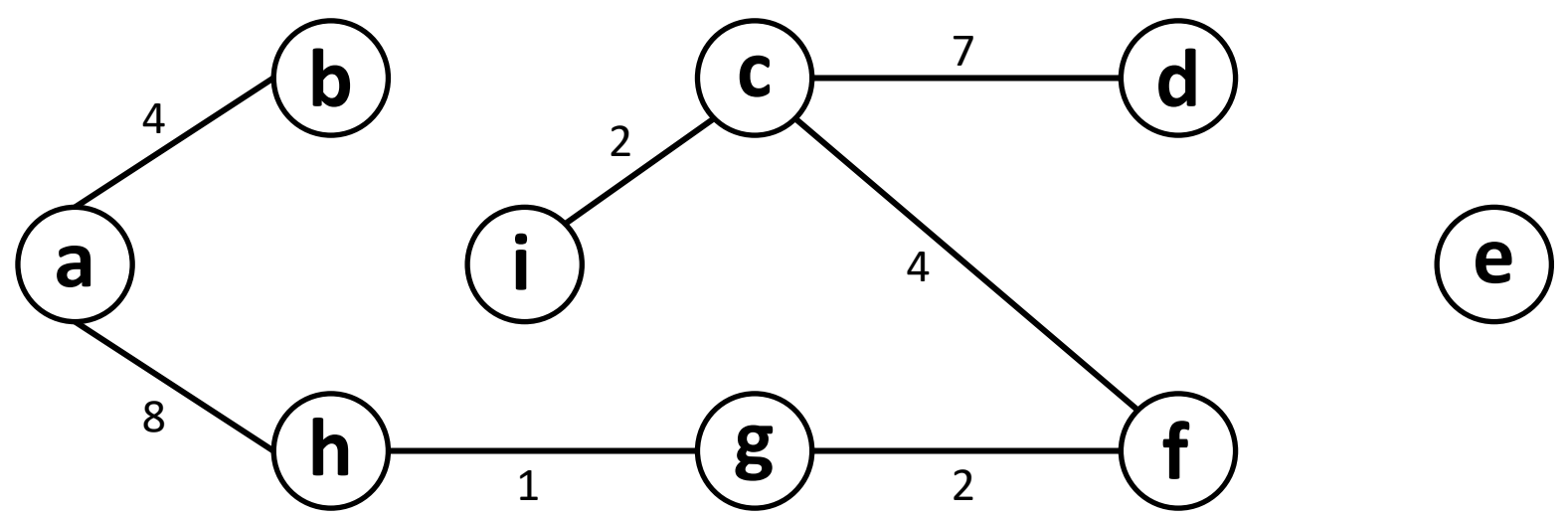


Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

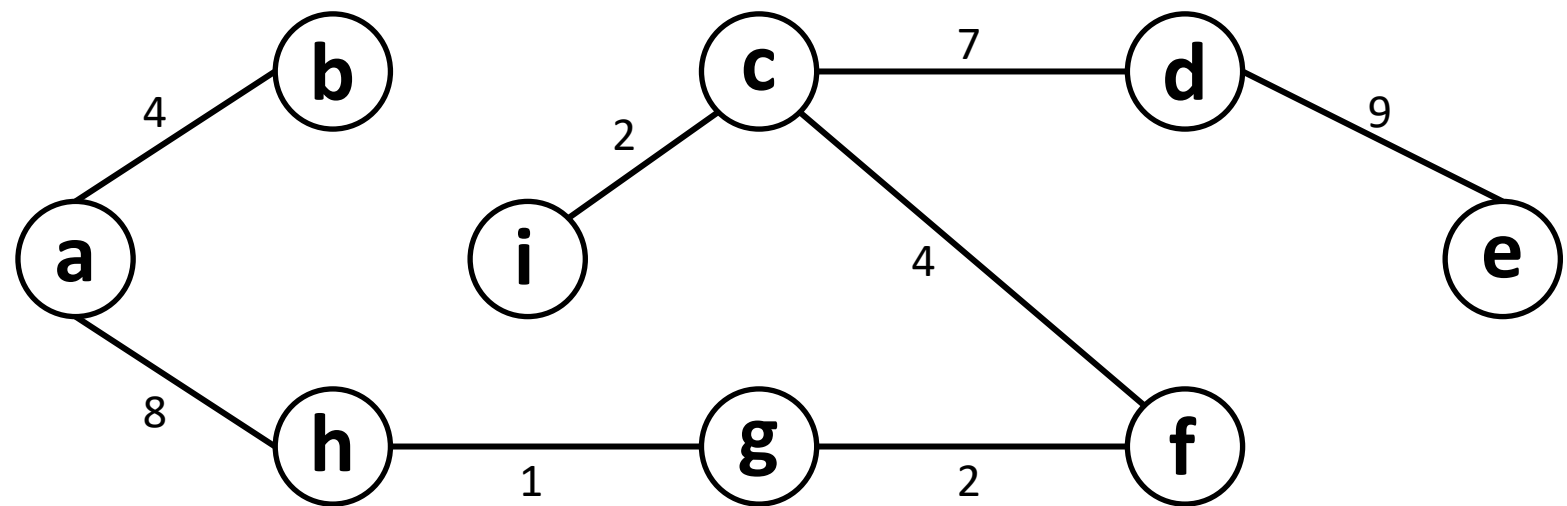


Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

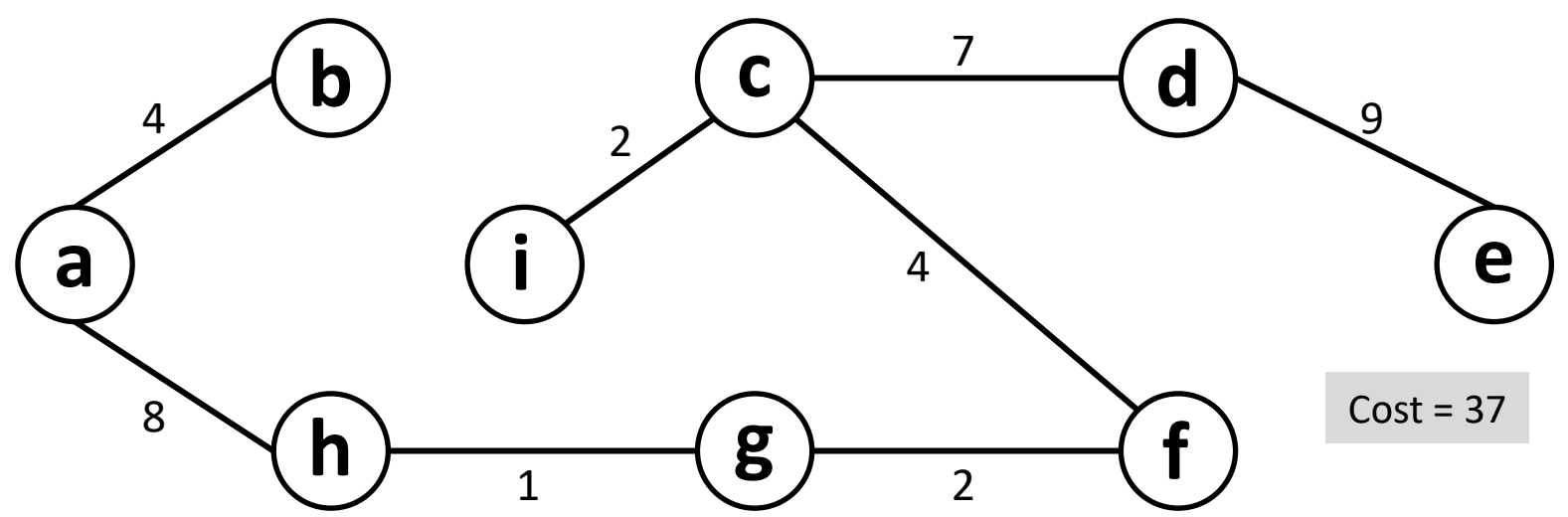


Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...



Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Implementation

MAKE-SET(x)

```
1   $x.p = x$   
2   $x.rank = 0$ 
```

MST-KRUSKAL(G, w)

```
1   $A = \emptyset$   
2  for each vertex  $v \in G.V$   
3      MAKE-SET( $v$ )  
4  sort the edges of  $G.E$  into nondecreasing order by weight  $w$   
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight  
6      if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ )  
7           $A = A \cup \{(u, v)\}$   
8          UNION( $u, v$ )  
9  return  $A$ 
```

FIND-SET(x)

```
1  if  $x \neq x.p$   
2       $x.p = \text{FIND-SET}(x.p)$   
3  return  $x.p$ 
```

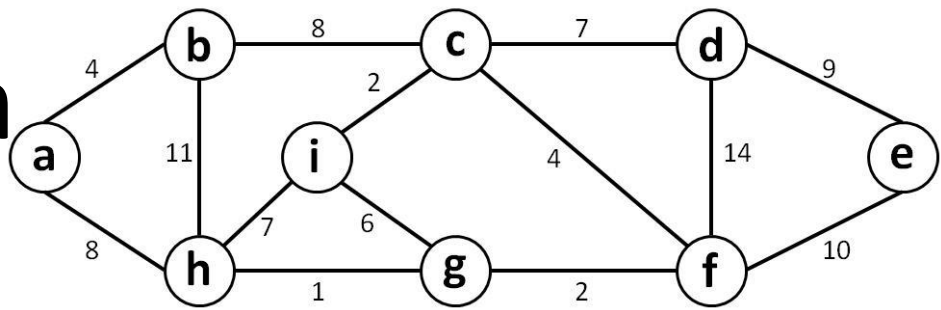
UNION(x, y)

```
1  LINK(FIND-SET( $x$ ), FIND-SET( $y$ ))
```

LINK(x, y)

```
1  if  $x.rank > y.rank$   
2       $y.p = x$   
3  else  $x.p = y$   
4      if  $x.rank == y.rank$   
5           $y.rank = y.rank + 1$ 
```

Example - Execution



Vertex	p	rank
a	a	0
b	b	0
c	c	0
d	d	0
e	e	0
f	f	0
g	g	0
h	h	0
i	i	0

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

A = {}

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	a	0
b	b	0
c	c	0
d	d	0
e	e	0
f	f	0
g	g	0
h	h	0
i	i	0

A = {}

```
FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
```

```
LINK(x, y)
1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
```

```
MST-KRUSKAL( $G, w$ )
1   $A = \emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET( $v$ )
4  sort the edges of  $G.E$  into nondecreasing order by weight  $w$ 
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ )
7           $A = A \cup \{(u, v)\}$ 
8          UNION( $u, v$ )
9  return  $A$ 
```

```
UNION(x, y)
1  LINK(FIND-SET( $x$ ), FIND-SET( $y$ ))
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	a	0
b	b	0
c	c	0
d	d	0
e	e	0
f	f	0
g	h	0
h	h	1
i	i	0

A = {(g,h)}

```
FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
```

```
LINK(x, y)
1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
```

```
MST-KRUSKAL(G, w)
1  A =  $\emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET(v)
4  sort the edges of  $G.E$  into nondecreasing order by weight w
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET(u)  $\neq$  FIND-SET(v)
7           $A = A \cup \{(u, v)\}$ 
8          UNION(u, v)
9  return A
```

```
UNION(x, y)
1  LINK(FIND-SET(x), FIND-SET(y))
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	a	0
b	b	0
c	c	0
d	d	0
e	e	0
f	f	0
g	h	0
h	h	1
i	i	0

A = {(g,h)}

```
FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
```

```
LINK(x, y)
1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
```

```
MST-KRUSKAL( $G, w$ )
1   $A = \emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET( $v$ )
4  sort the edges of  $G.E$  into nondecreasing order by weight  $w$ 
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ )
7           $A = A \cup \{(u, v)\}$ 
8          UNION( $u, v$ )
9  return  $A$ 
```

```
UNION(x, y)
1  LINK(FIND-SET( $x$ ), FIND-SET( $y$ ))
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	a	0
b	b	0
c	c	0
d	d	0
e	e	0
f	h	0
g	h	0
h	h	1
i	i	0

$A = \{(g,h), (f,g)\}$

```
FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
```

```
LINK(x, y)
1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
```

```
MST-KRUSKAL( $G, w$ )
1   $A = \emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET( $v$ )
4  sort the edges of  $G.E$  into nondecreasing order by weight  $w$ 
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ )
7           $A = A \cup \{(u, v)\}$ 
8          UNION( $u, v$ )
9  return  $A$ 
```

```
UNION(x, y)
1  LINK(FIND-SET( $x$ ), FIND-SET( $y$ ))
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	a	0
b	b	0
c	c	0
d	d	0
e	e	0
f	h	0
g	h	0
h	h	1
i	i	0

A = {(g,h), (f,g)}

```
FIND-SET(x)
1  if x ≠ x.p
2      x.p = FIND-SET(x.p)
3  return x.p
```

```
LINK(x, y)
1  if x.rank > y.rank
2      y.p = x
3  else x.p = y
4      if x.rank == y.rank
5          y.rank = y.rank + 1
```

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

```
UNION(x, y)
1  LINK(FIND-SET(x), FIND-SET(y))
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	a	0
b	b	0
c	i	0
d	d	0
e	e	0
f	h	0
g	h	0
h	h	1
i	i	1

$A = \{(g,h), (f,g), (c,i)\}$

```
FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
```

```
LINK(x, y)
1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
```

```
MST-KRUSKAL( $G, w$ )
1   $A = \emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET( $v$ )
4  sort the edges of  $G.E$  into nondecreasing order by weight  $w$ 
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ )
7           $A = A \cup \{(u, v)\}$ 
8          UNION( $u, v$ )
9  return  $A$ 
```

```
UNION(x, y)
1  LINK(FIND-SET( $x$ ), FIND-SET( $y$ ))
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	a	0
b	b	0
c	i	0
d	d	0
e	e	0
f	h	0
g	h	0
h	h	1
i	i	1

A = {(g,h), (f,g), (c,i)}

```
FIND-SET(x)
1  if x ≠ x.p
2      x.p = FIND-SET(x.p)
3  return x.p
```

```
LINK(x, y)
1  if x.rank > y.rank
2      y.p = x
3  else x.p = y
4      if x.rank == y.rank
5          y.rank = y.rank + 1
```

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

```
UNION(x, y)
1  LINK(FIND-SET(x), FIND-SET(y))
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	b	0
b	b	1
c	i	0
d	d	0
e	e	0
f	h	0
g	h	0
h	h	1
i	i	1

A = {(g,h), (f,g), (c,i), (a,b)}

```
FIND-SET(x)
1  if x ≠ x.p
2      x.p = FIND-SET(x.p)
3  return x.p
```

```
LINK(x, y)
1  if x.rank > y.rank
2      y.p = x
3  else x.p = y
4      if x.rank == y.rank
5          y.rank = y.rank + 1
```

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

```
UNION(x, y)
1  LINK(FIND-SET(x), FIND-SET(y))
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	b	0
b	b	1
c	i	0
d	d	0
e	e	0
f	h	0
g	h	0
h	h	1
i	i	1

A = {(g,h), (f,g), (c,i), (a,b)}

```
FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
```

```
LINK(x, y)
1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
```

```
MST-KRUSKAL(G, w)
1  A =  $\emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET(v)
4  sort the edges of  $G.E$  into nondecreasing order by weight w
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET(u)  $\neq$  FIND-SET(v)
7           $A = A \cup \{(u, v)\}$ 
8          UNION(u, v)
9  return A
```

```
UNION(x, y)
1  LINK(FIND-SET(x), FIND-SET(y))
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	b	0
b	b	1
c	i	0
d	d	0
e	e	0
f	h	0
g	h	0
h	h	2
i	h	1

A = {(g,h), (f,g), (c,i),
(a,b), (c,f)}

```
FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
```

```
LINK(x, y)
1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
```

```
MST-KRUSKAL(G, w)
1  A =  $\emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET(v)
4  sort the edges of  $G.E$  into nondecreasing order by weight w
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET(u)  $\neq$  FIND-SET(v)
7           $A = A \cup \{(u, v)\}$ 
8          UNION(u, v)
9  return A
```

```
UNION(x, y)
1  LINK(FIND-SET(x), FIND-SET(y))
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	b	0
b	b	1
c	i	0
d	d	0
e	e	0
f	h	0
g	h	0
h	h	2
i	h	1

A = {(g,h), (f,g), (c,i),
(a,b), (c,f)}

```
FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
```

```
LINK(x, y)
1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
```

```
MST-KRUSKAL(G, w)
1  A =  $\emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET(v)
4  sort the edges of  $G.E$  into nondecreasing order by weight w
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET(u)  $\neq$  FIND-SET(v)
7           $A = A \cup \{(u, v)\}$ 
8          UNION(u, v)
9  return A
```

```
UNION(x, y)
1  LINK(FIND-SET(x), FIND-SET(y))
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	b	0
b	b	1
c	i	0
d	d	0
e	e	0
f	h	0
g	h	0
h	h	2
i	h	1

$A = \{(g,h), (f,g), (c,i), (a,b), (c,f)\}$

```

FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
    
```

LINK(x, y)

```

1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
    
```

MST-KRUSKAL(G, w)

```

1   $A = \emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET( $v$ )
4  sort the edges of  $G.E$  into nondecreasing order by weight  $w$ 
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ )
7           $A = A \cup \{(u, v)\}$ 
8          UNION( $u, v$ )
9  return  $A$ 
    
```

UNION(x, y)

```

1  LINK(FIND-SET( $x$ ), FIND-SET( $y$ ))
    
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	b	0
b	b	1
c	h	0
d	h	0
e	e	0
f	h	0
g	h	0
h	h	2
i	h	1

A = {(g,h), (f,g), (c,i),
(a,b), (c,f), (c,d)}

```
FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
```

```
LINK(x, y)
1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
```

```
MST-KRUSKAL(G, w)
1  A =  $\emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET(v)
4  sort the edges of  $G.E$  into nondecreasing order by weight w
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET(u)  $\neq$  FIND-SET(v)
7           $A = A \cup \{(u, v)\}$ 
8          UNION(u, v)
9  return A
```

```
UNION(x, y)
1  LINK(FIND-SET(x), FIND-SET(y))
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	b	0
b	b	1
c	h	0
d	h	0
e	e	0
f	h	0
g	h	0
h	h	2
i	h	1

$A = \{(g,h), (f,g), (c,i), (a,b), (c,f), (c,d)\}$

```

FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
    
```

MST-KRUSKAL(G, w)

```

1   $A = \emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET( $v$ )
4  sort the edges of  $G.E$  into nondecreasing order by weight  $w$ 
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if  $\text{FIND-SET}(u) \neq \text{FIND-SET}(v)$ 
7           $A = A \cup \{(u, v)\}$ 
8          UNION( $u, v$ )
9  return  $A$ 
    
```

LINK(x, y)

```

1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
    
```

UNION(x, y)

```

1  LINK( $\text{FIND-SET}(x), \text{FIND-SET}(y)$ )
    
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	b	0
b	b	1
c	h	0
d	h	0
e	e	0
f	h	0
g	h	0
h	h	2
i	h	1

$A = \{(g,h), (f,g), (c,i), (a,b), (c,f), (c,d)\}$

```

FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
    
```

LINK(x, y)

```

1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
    
```

MST-KRUSKAL(G, w)

```

1   $A = \emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET( $v$ )
4  sort the edges of  $G.E$  into nondecreasing order by weight  $w$ 
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ )
7           $A = A \cup \{(u, v)\}$ 
8          UNION( $u, v$ )
9  return  $A$ 
    
```

UNION(x, y)

```

1  LINK(FIND-SET( $x$ ), FIND-SET( $y$ ))
    
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	b	0
b	h	1
c	h	0
d	h	0
e	e	0
f	h	0
g	h	0
h	h	2
i	h	1

$A = \{(g,h), (f,g), (c,i), (a,b), (c,f), (c,d), (a,h)\}$

```

FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
    
```

LINK(x, y)

```

1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
    
```

MST-KRUSKAL(G, w)

```

1   $A = \emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET( $v$ )
4  sort the edges of  $G.E$  into nondecreasing order by weight  $w$ 
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ )
7           $A = A \cup \{(u, v)\}$ 
8          UNION( $u, v$ )
9  return  $A$ 
    
```

UNION(x, y)

```

1  LINK(FIND-SET( $x$ ), FIND-SET( $y$ ))
    
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	b	0
b	h	1
c	h	0
d	h	0
e	e	0
f	h	0
g	h	0
h	h	2
i	h	1

$A = \{(g,h), (f,g), (c,i), (a,b), (c,f), (c,d), (a,h)\}$

```

FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
    
```

LINK(x, y)

```

1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
    
```

MST-KRUSKAL(G, w)

```

1   $A = \emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET( $v$ )
4  sort the edges of  $G.E$  into nondecreasing order by weight  $w$ 
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ )
7           $A = A \cup \{(u, v)\}$ 
8          UNION( $u, v$ )
9  return  $A$ 
    
```

UNION(x, y)

```

1  LINK(FIND-SET( $x$ ), FIND-SET( $y$ ))
    
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	b	0
b	h	1
c	h	0
d	h	0
e	e	0
f	h	0
g	h	0
h	h	2
i	h	1

$A = \{(g,h), (f,g), (c,i),$
 $(a,b), (c,f), (c,d),$
 $(a,h)\}$

```

FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
    
```

LINK(x, y)

```

1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
    
```

MST-KRUSKAL(G, w)

```

1   $A = \emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET( $v$ )
4  sort the edges of  $G.E$  into nondecreasing order by weight  $w$ 
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ )
7           $A = A \cup \{(u, v)\}$ 
8          UNION( $u, v$ )
9  return  $A$ 
    
```

UNION(x, y)

```

1  LINK(FIND-SET( $x$ ), FIND-SET( $y$ ))
    
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	b	0
b	h	1
c	h	0
d	h	0
e	h	0
f	h	0
g	h	0
h	h	2
i	h	1

A = {(g,h), (f,g), (c,i),
(a,b), (c,f), (c,d),
(a,h), (d,e)}

```
FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
```

```
LINK(x, y)
1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
```

```
MST-KRUSKAL(G, w)
1  A =  $\emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET(v)
4  sort the edges of  $G.E$  into nondecreasing order by weight w
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET(u)  $\neq$  FIND-SET(v)
7           $A = A \cup \{(u, v)\}$ 
8          UNION(u, v)
9  return A
```

```
UNION(x, y)
1  LINK(FIND-SET(x), FIND-SET(y))
```

Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

Contd...

Vertex	p	rank
a	b	0
b	h	1
c	h	0
d	h	0
e	h	0
f	h	0
g	h	0
h	h	2
i	h	1

$A = \{(g,h), (f,g), (c,i),$
 $(a,b), (c,f), (c,d),$
 $(a,h), (d,e)\}$

```

FIND-SET(x)
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
    
```

LINK(x, y)

```

1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else  $x.p = y$ 
4      if  $x.rank == y.rank$ 
5           $y.rank = y.rank + 1$ 
    
```

MST-KRUSKAL(G, w)

```

1   $A = \emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET( $v$ )
4  sort the edges of  $G.E$  into nondecreasing order by weight  $w$ 
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ )
7           $A = A \cup \{(u, v)\}$ 
8          UNION( $u, v$ )
9  return  $A$ 
    
```

UNION(x, y)

```

1  LINK(FIND-SET( $x$ ), FIND-SET( $y$ ))
    
```

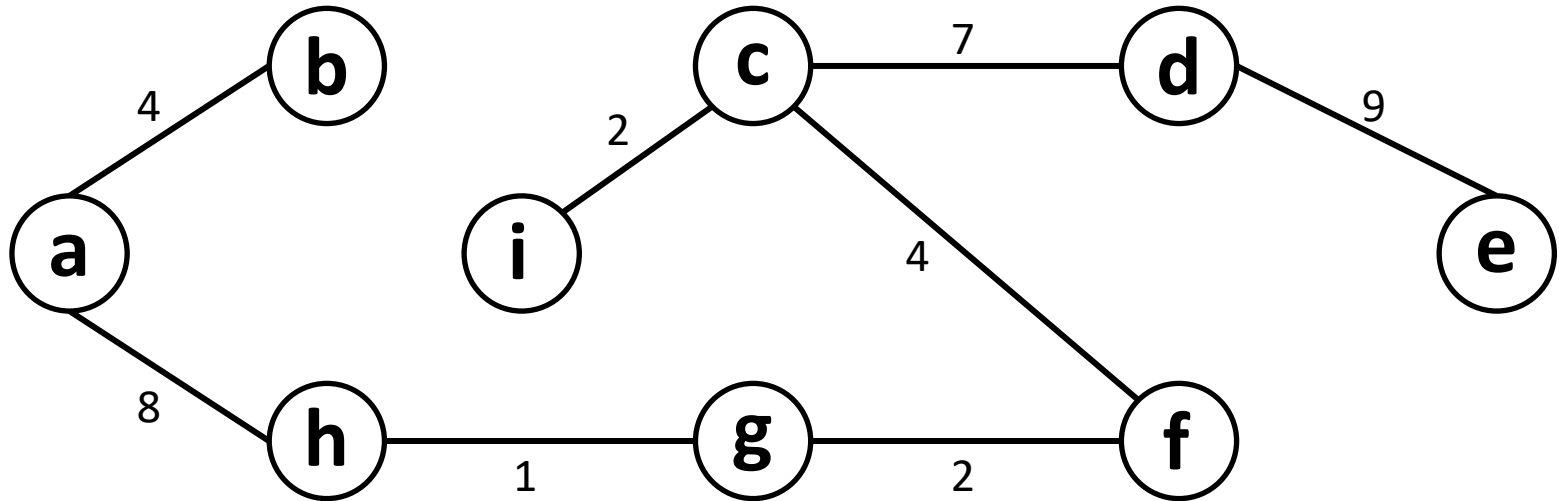
Edge	Weight
(g,h)	1
(f,g)	2
(c,i)	2
(a,b)	4
(c,f)	4

Edge	Weight
(g,i)	6
(c,d)	7
(h,i)	7
(a,h)	8
(b,c)	8

Edge	Weight
(d,e)	9
(e,f)	10
(b,h)	11
(d,f)	14

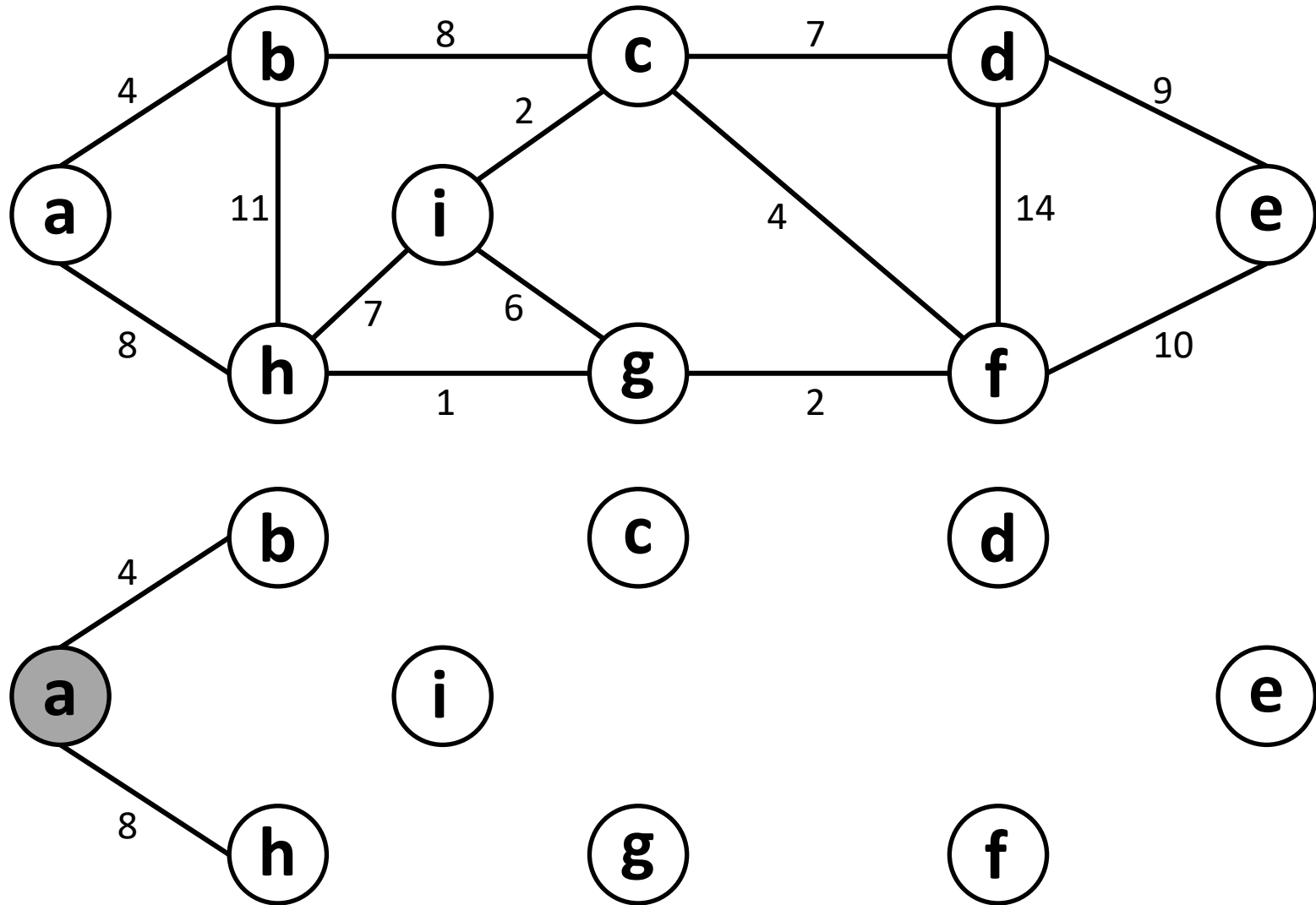
Contd...

$A = \{(g,h), (f,g), (c,i), (a,b), (c,f), (c,d), (a,h), (d,e)\}$

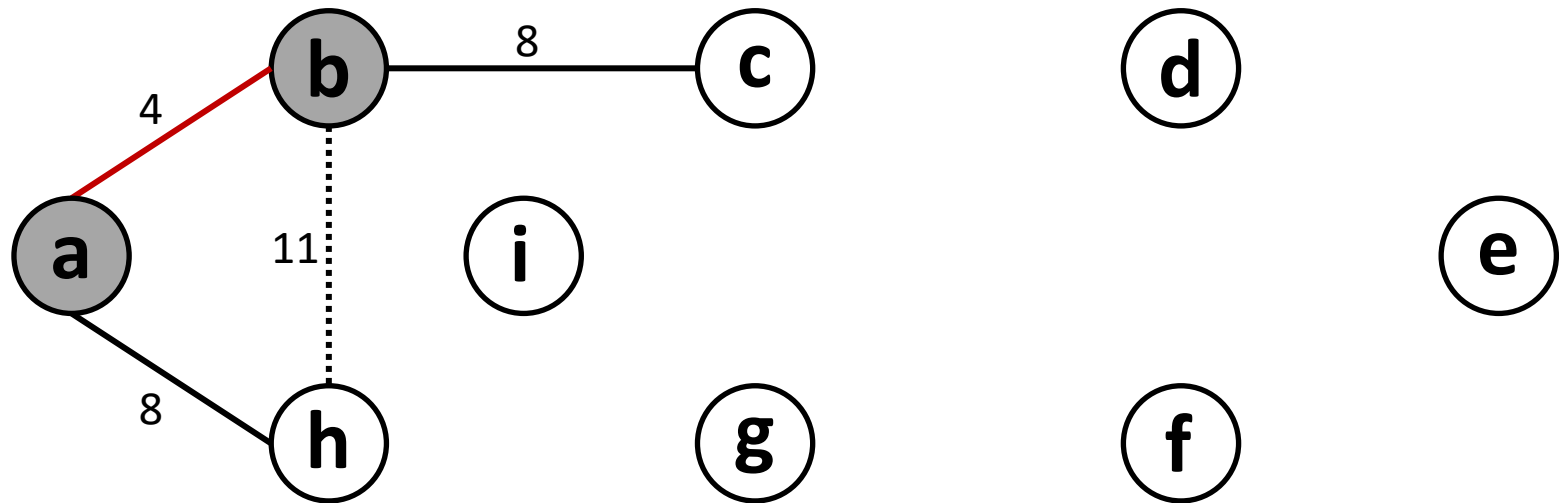
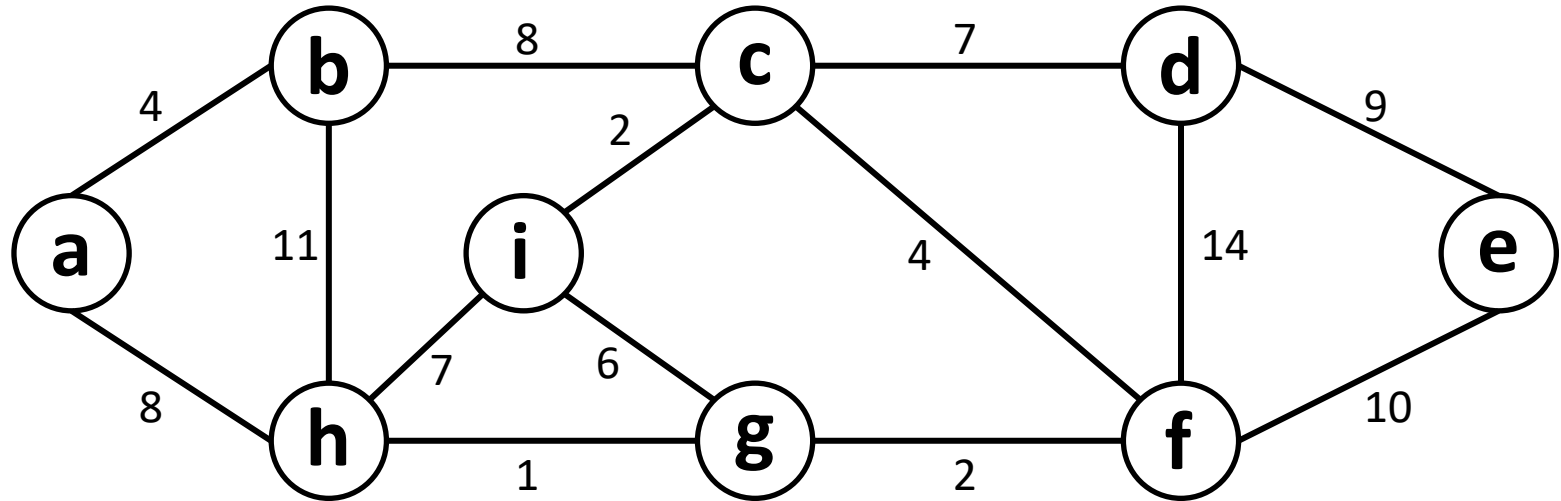


Cost = 37

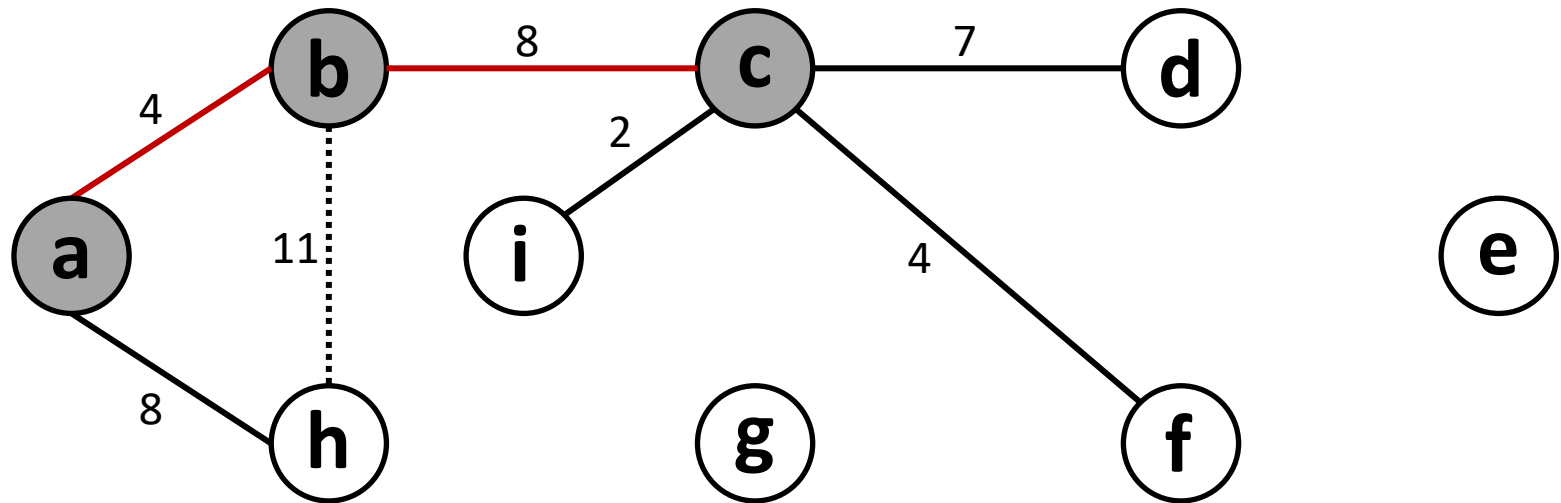
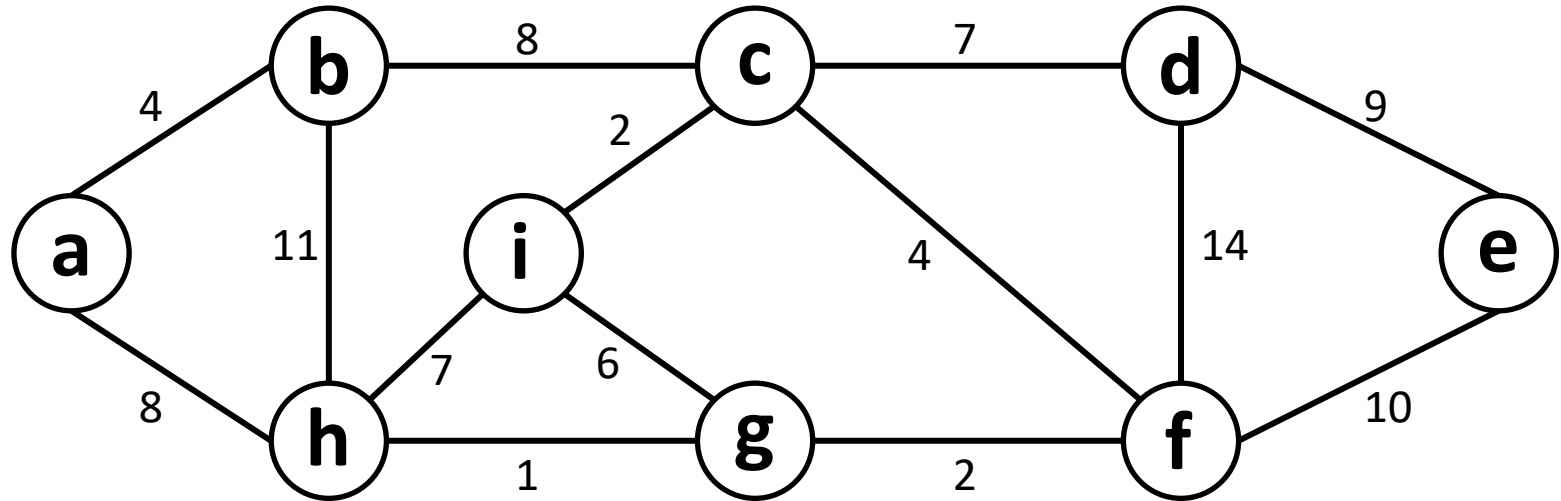
Prim's Algorithm



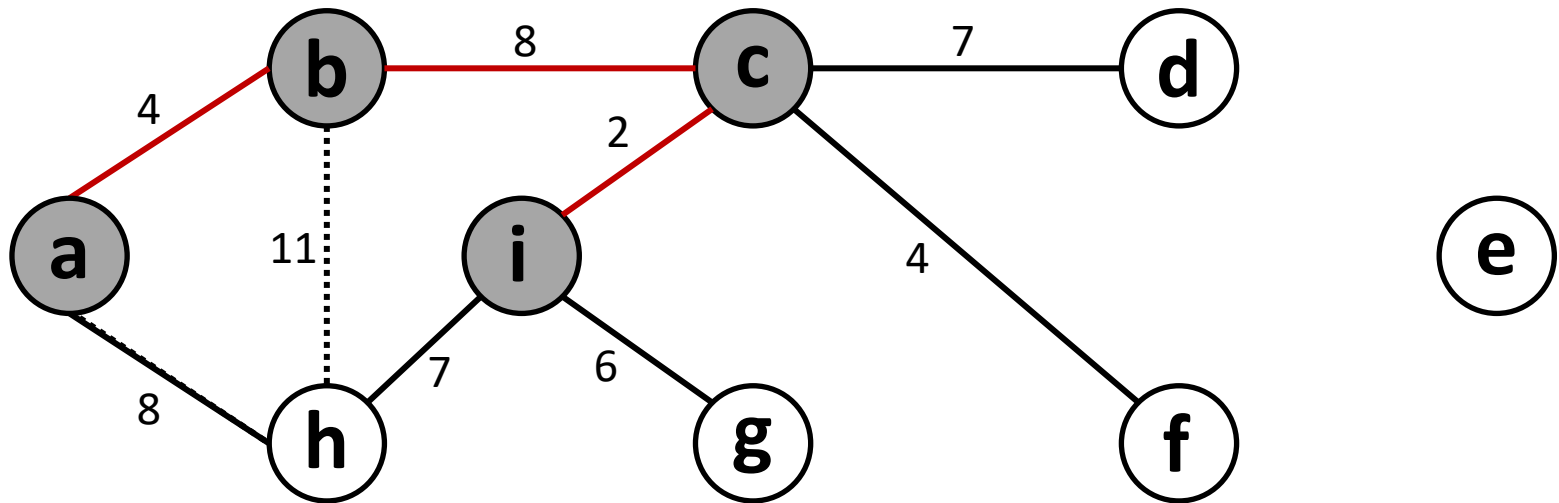
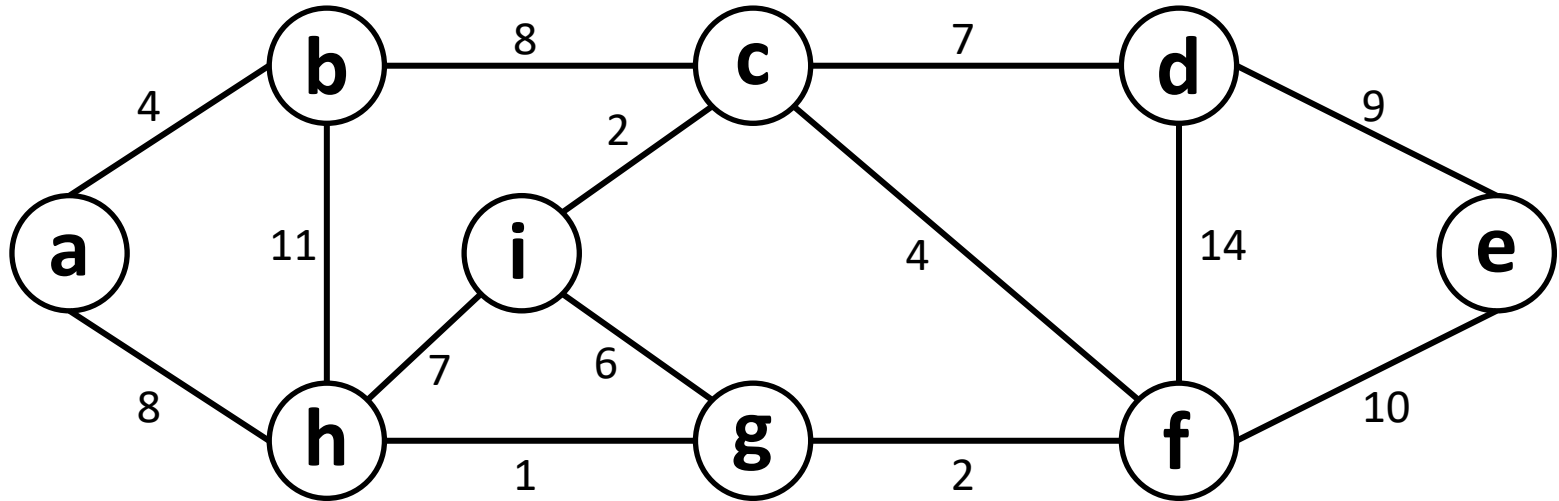
Prim's Algorithm



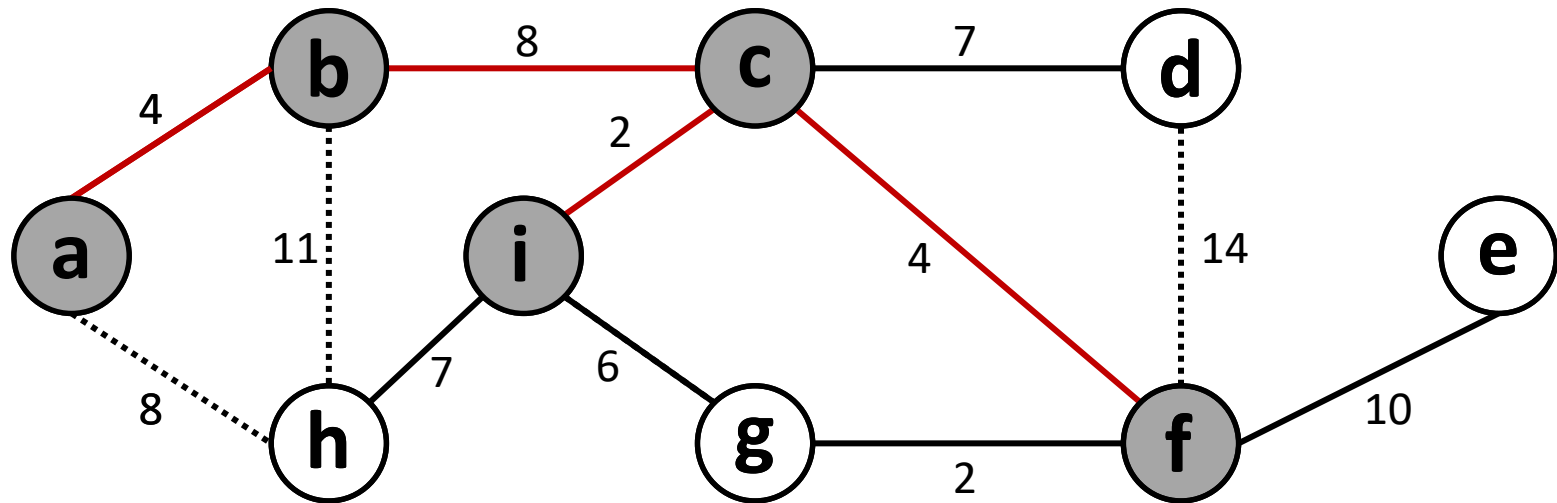
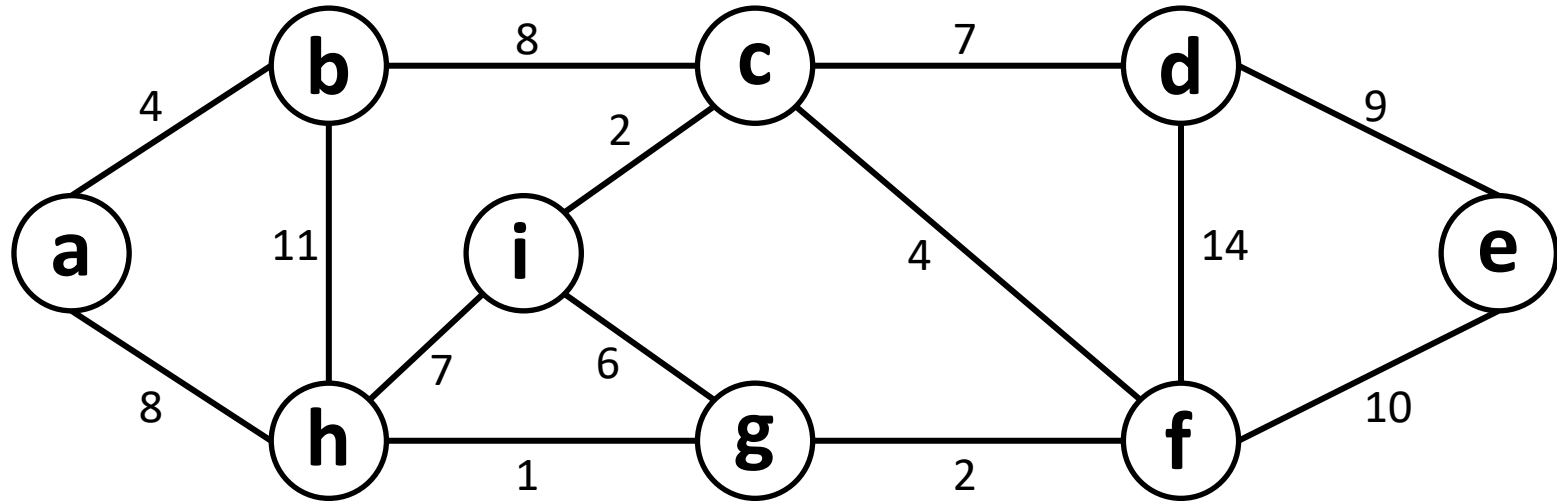
Prim's Algorithm



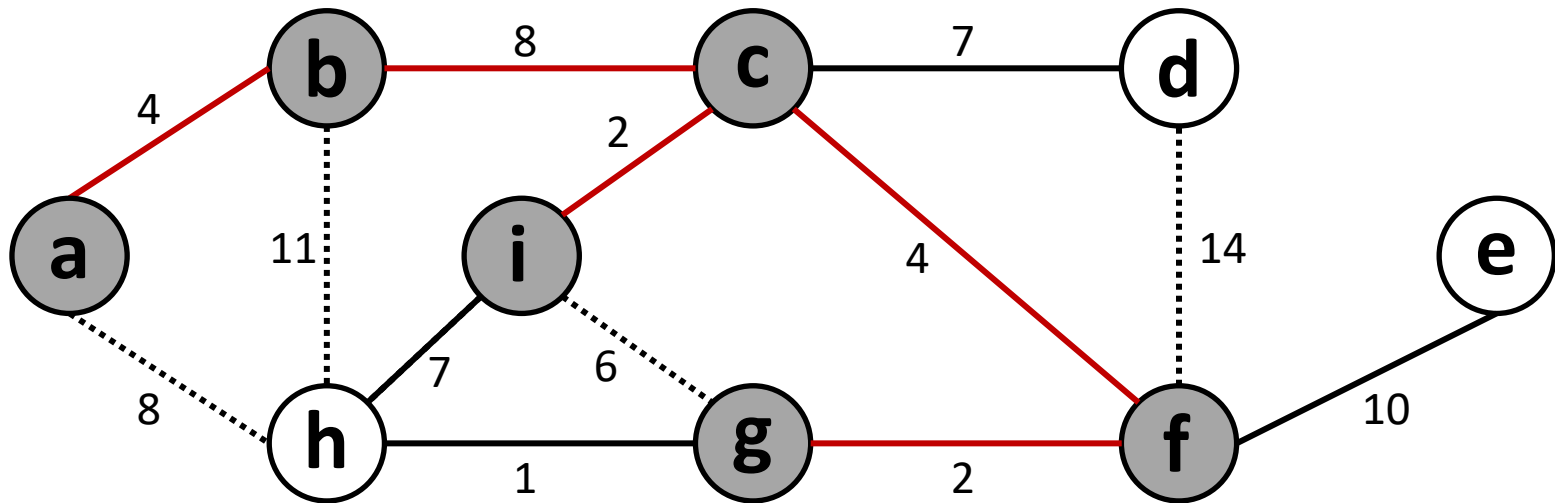
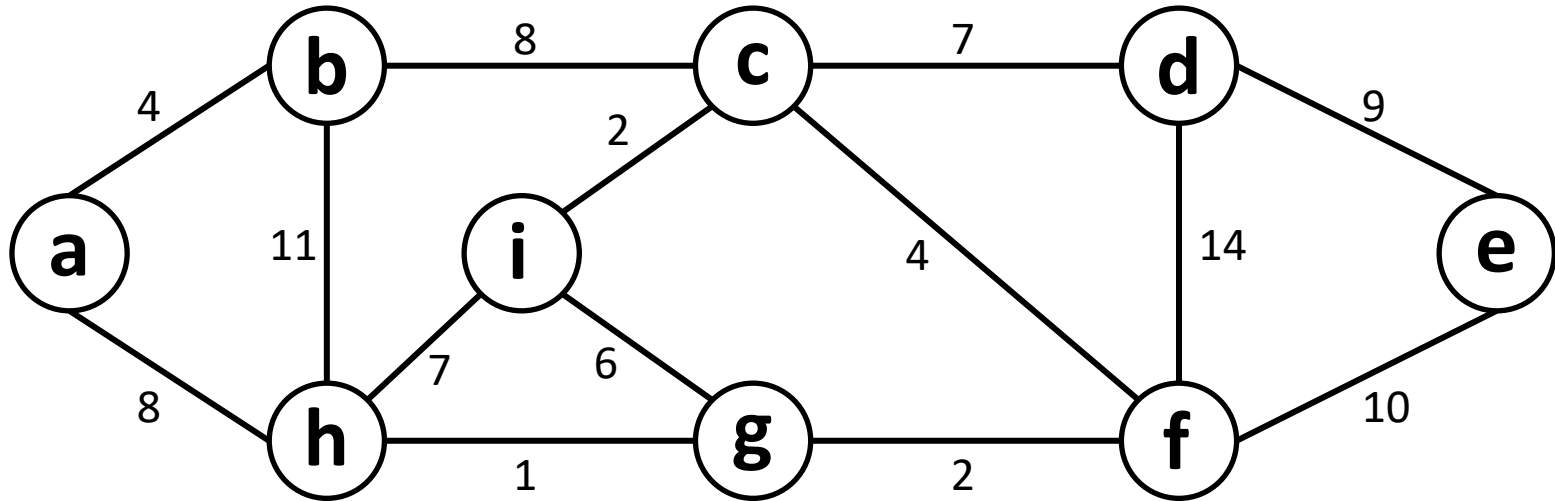
Prim's Algorithm



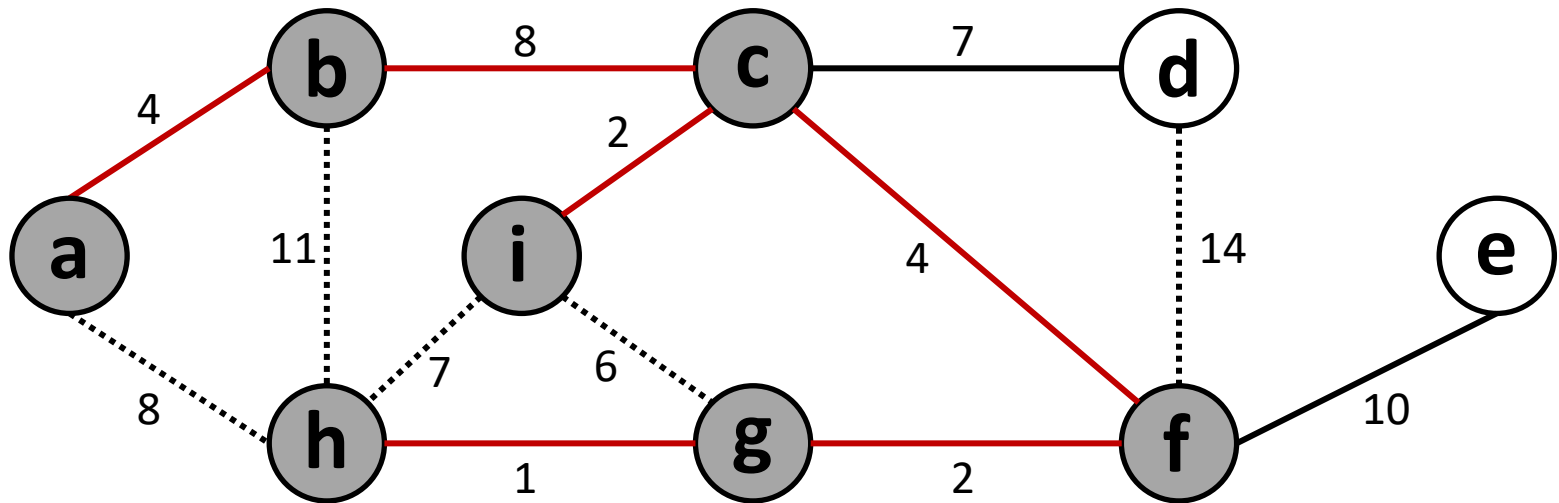
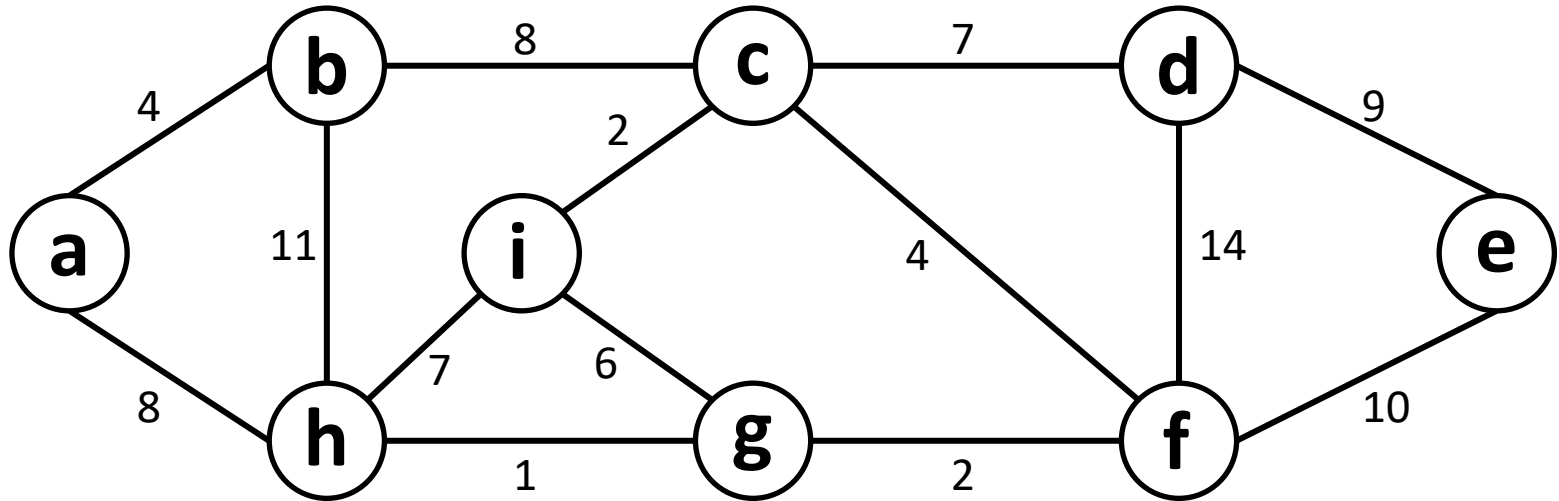
Prim's Algorithm



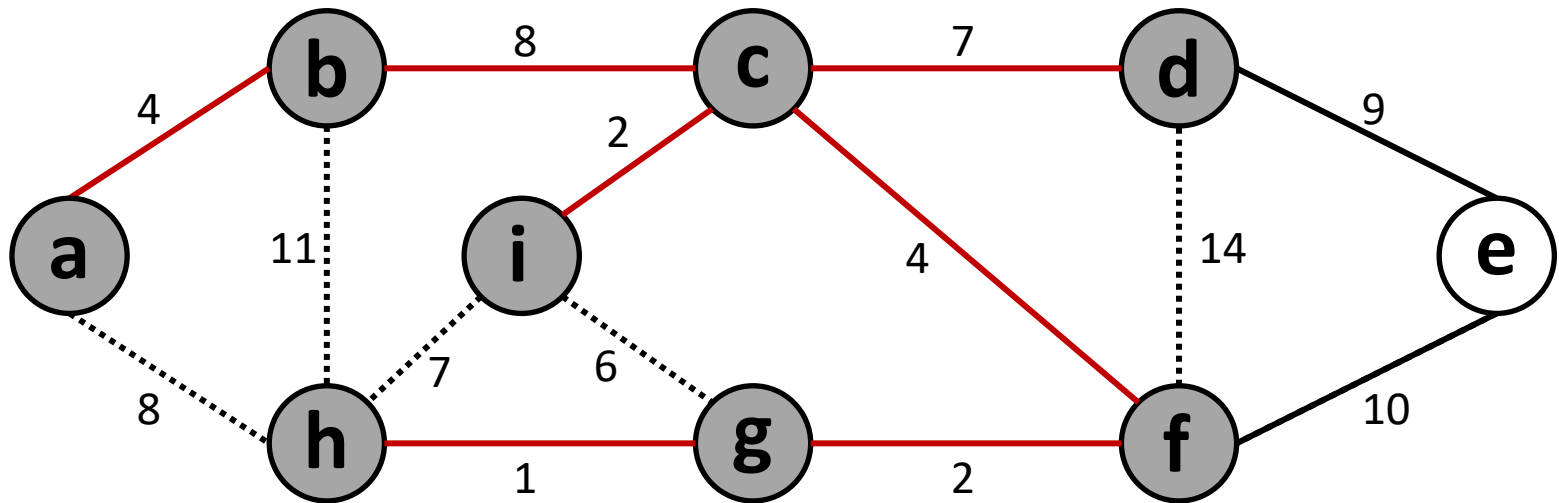
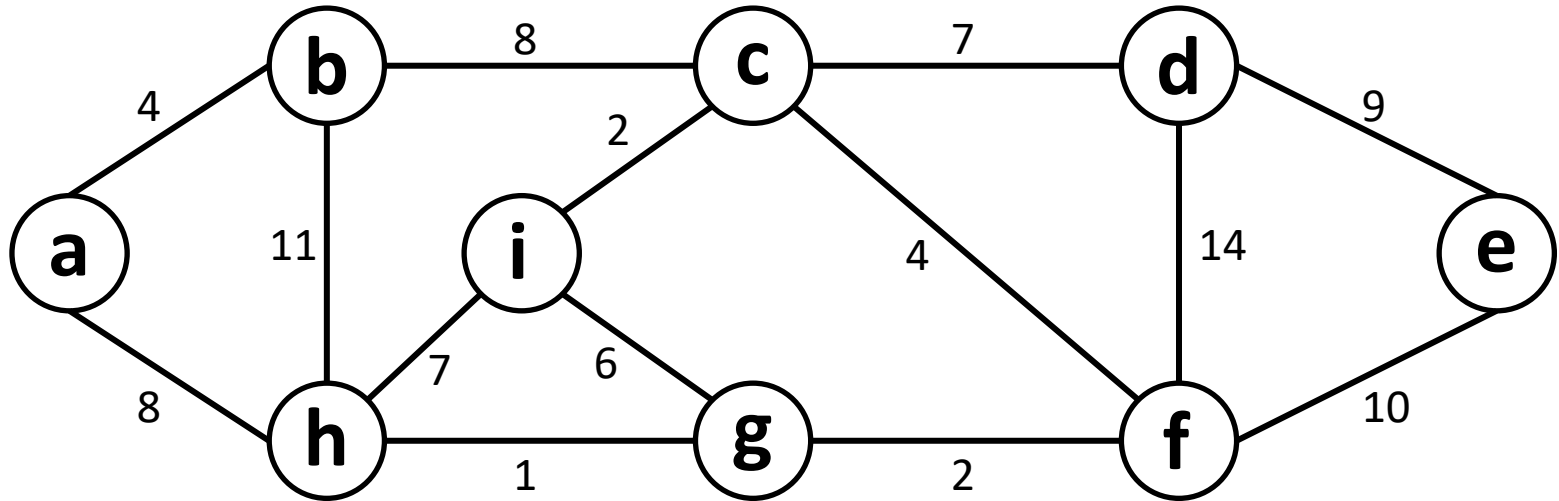
Prim's Algorithm



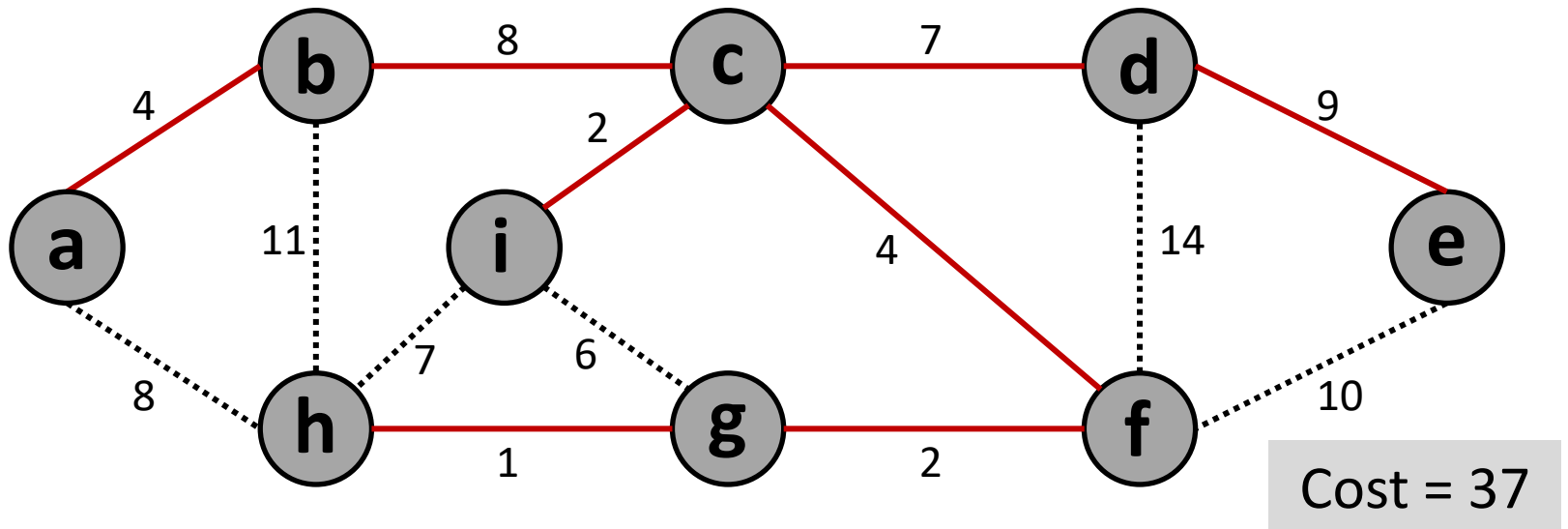
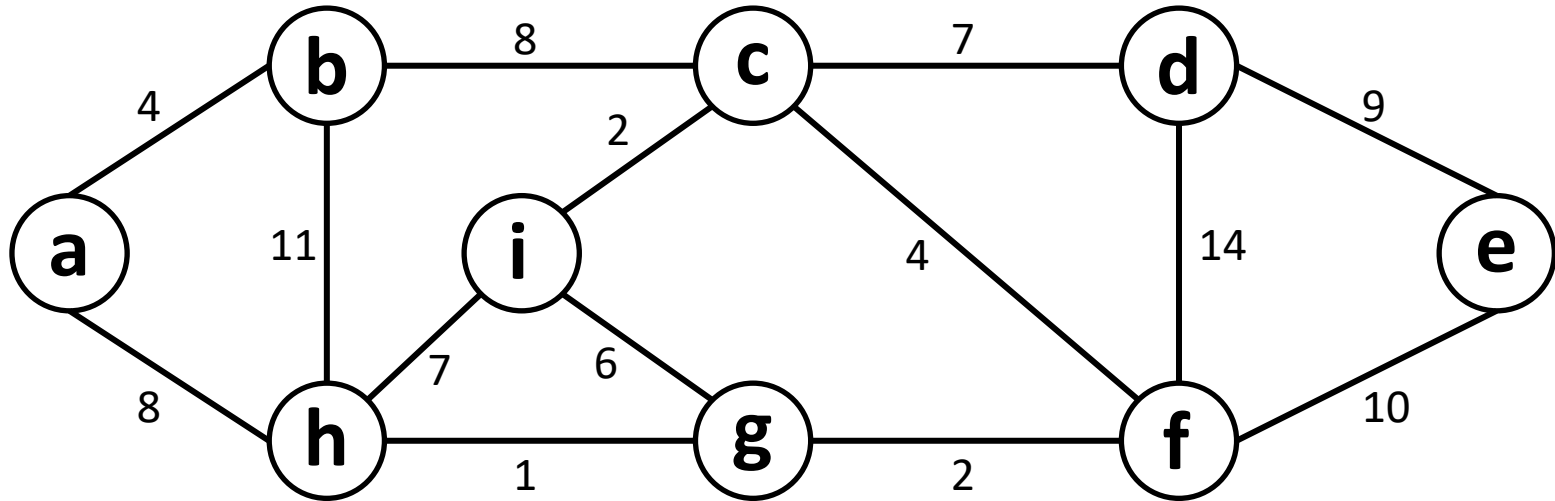
Prim's Algorithm



Prim's Algorithm



Prim's Algorithm



Implementation

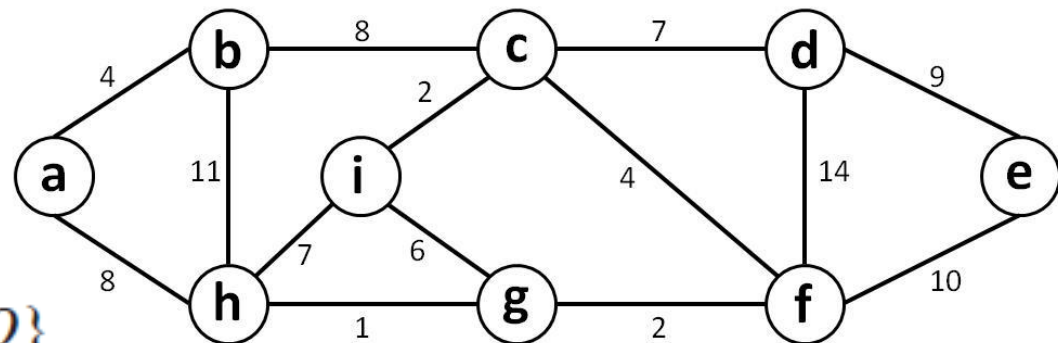
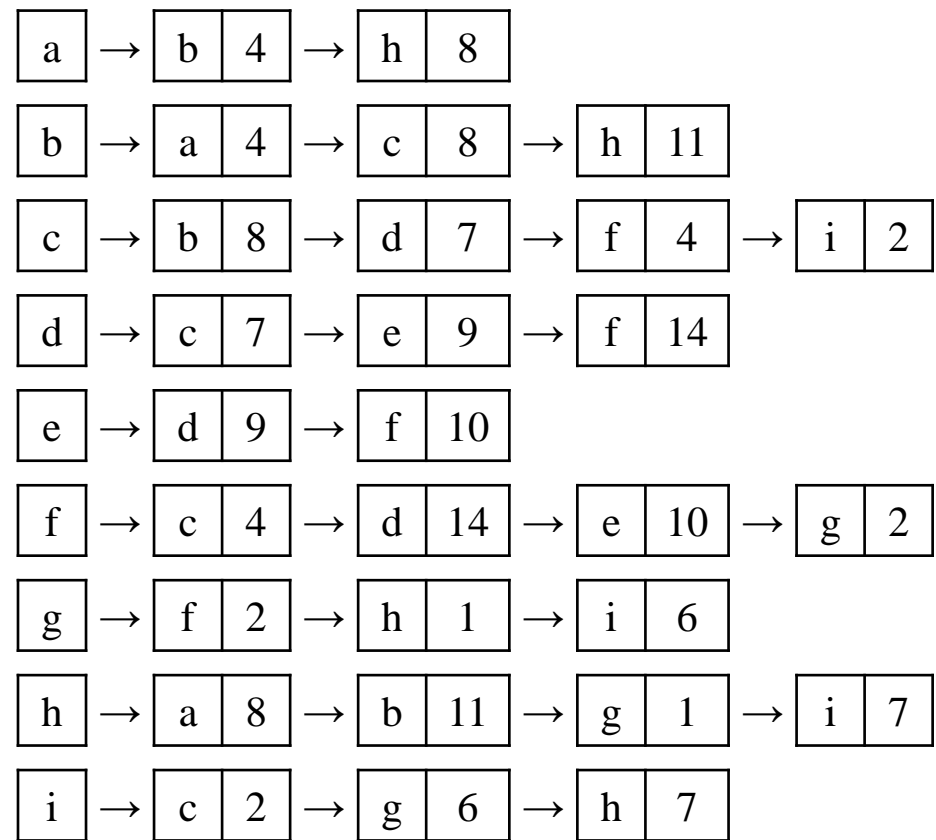
MST-PRIM(G, w, r)

```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 

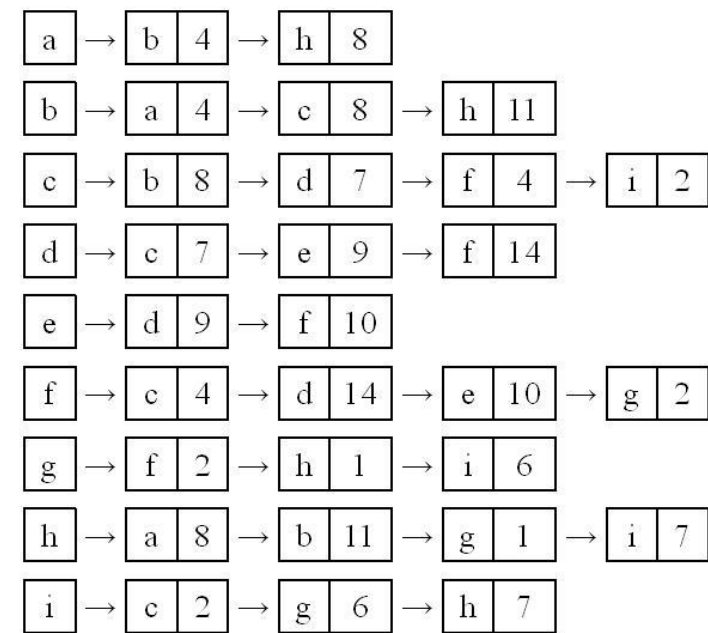
```

$A = \{(v, v.\pi) : v \in V - \{r\} - Q\}$



Example - Execution

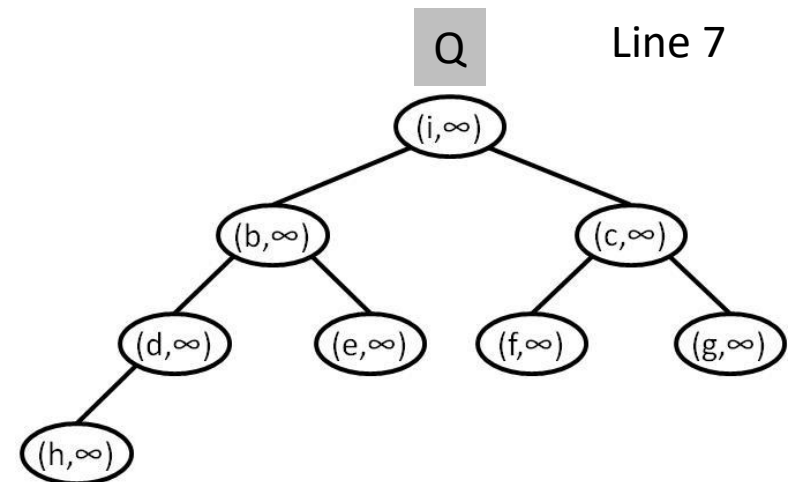
Vertex	π	key
a	NIL	0
b	NIL	∞
c	NIL	∞
d	NIL	∞
e	NIL	∞
f	NIL	∞
g	NIL	∞
h	NIL	∞
i	NIL	∞



MST-PRIM(G, w, r)

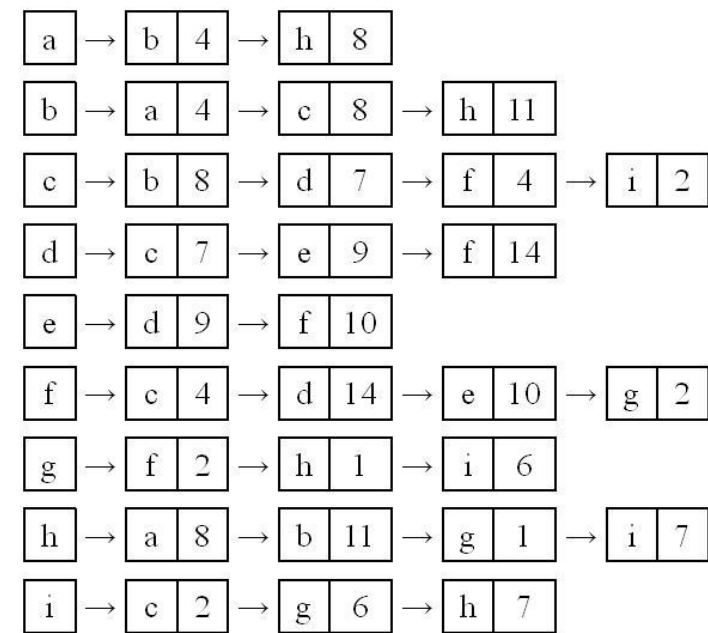
```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
  
```



Contd...

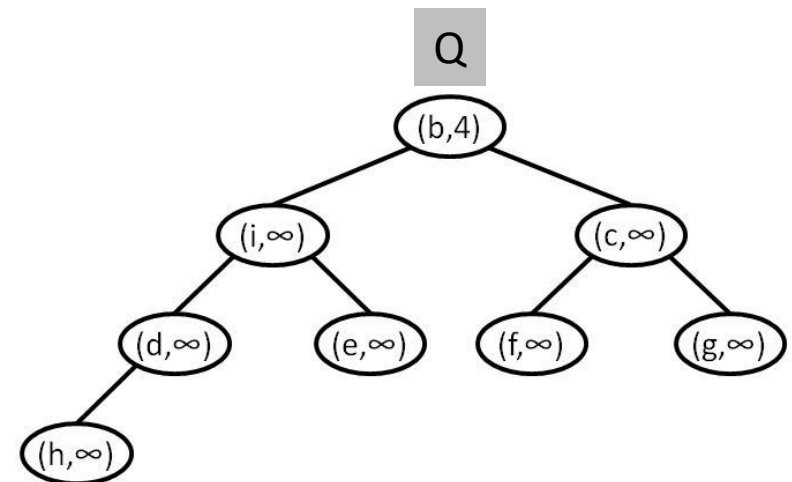
Vertex	π	key
a	NIL	0
b	a	4
c	NIL	∞
d	NIL	∞
e	NIL	∞
f	NIL	∞
g	NIL	∞
h	NIL	∞
i	NIL	∞



MST-PRIM(G, w, r)

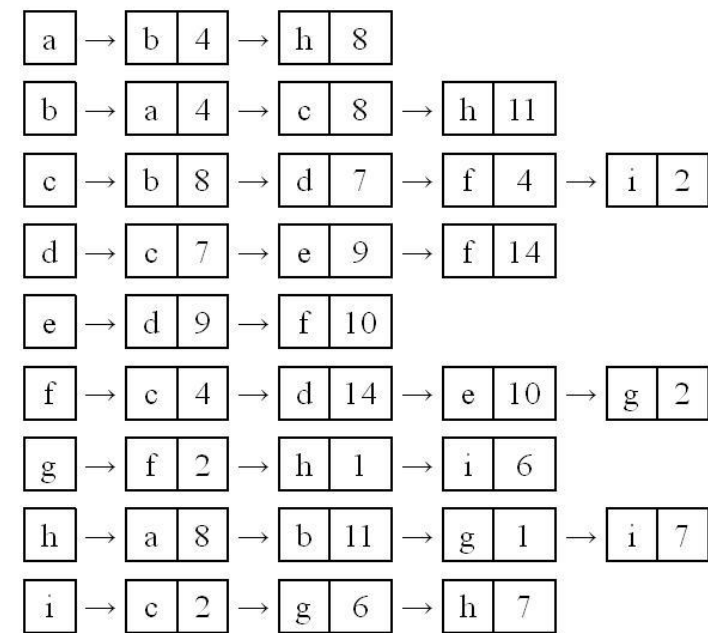
```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
  
```



Contd...

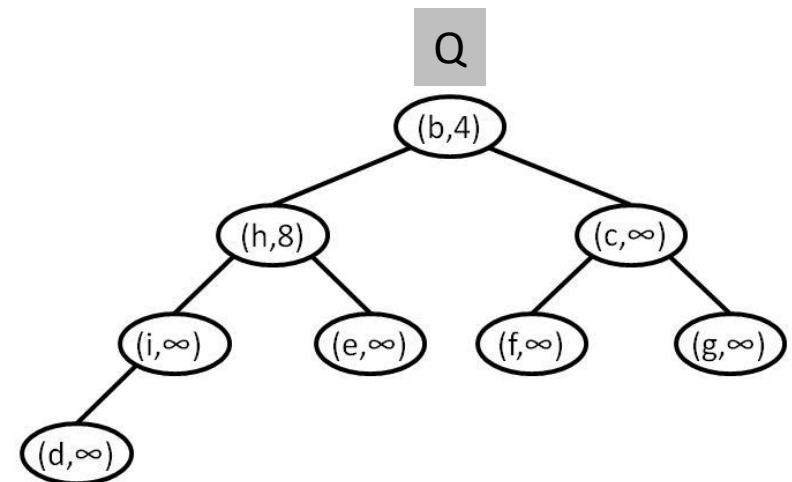
Vertex	π	key
a	NIL	0
b	a	4
c	NIL	∞
d	NIL	∞
e	NIL	∞
f	NIL	∞
g	NIL	∞
h	a	8
i	NIL	∞



MST-PRIM(G, w, r)

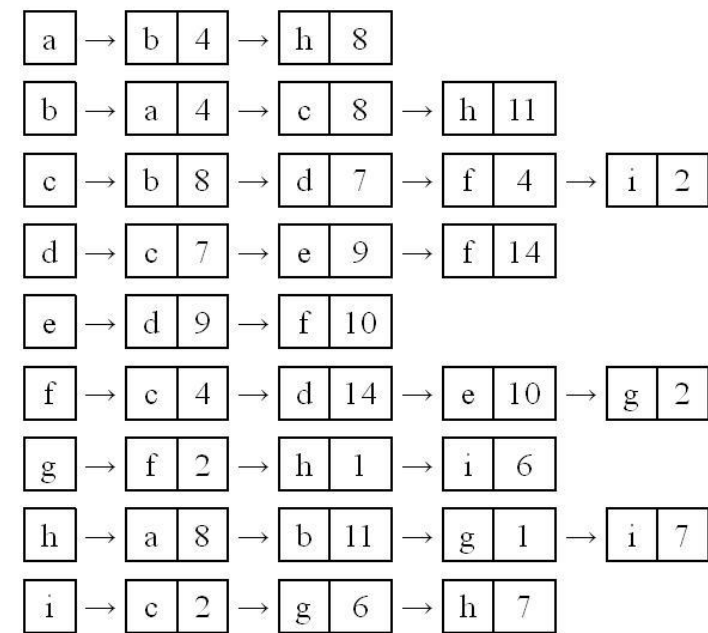
```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
  
```



Contd...

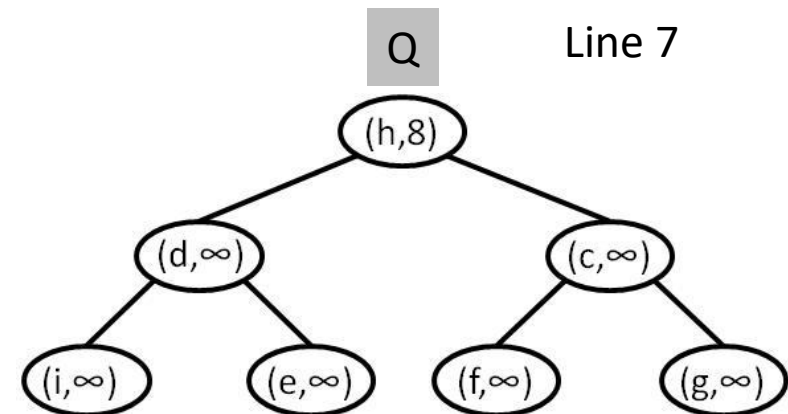
Vertex	π	key
a	NIL	0
b	a	4
c	NIL	∞
d	NIL	∞
e	NIL	∞
f	NIL	∞
g	NIL	∞
h	a	8
i	NIL	∞



MST-PRIM(G, w, r)

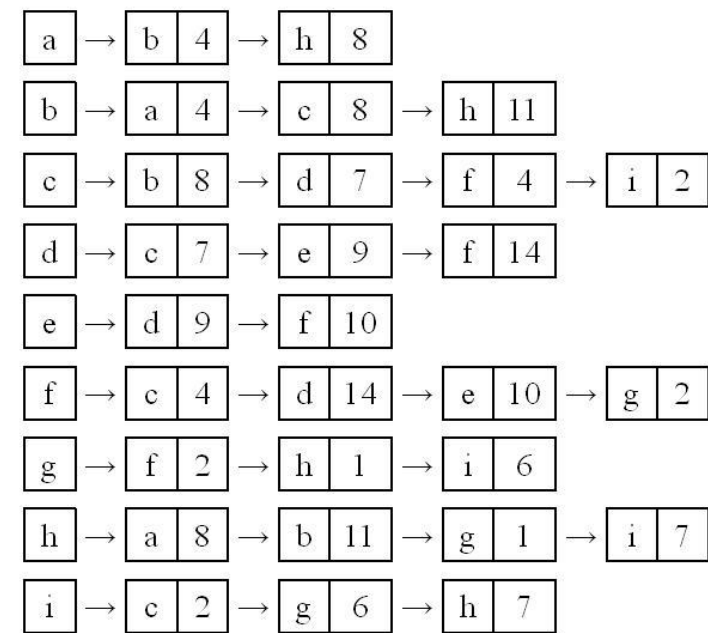
```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
  
```



Contd...

Vertex	π	key
a	NIL	0
b	a	4
c	b	8
d	NIL	∞
e	NIL	∞
f	NIL	∞
g	NIL	∞
h	a	8
i	NIL	∞

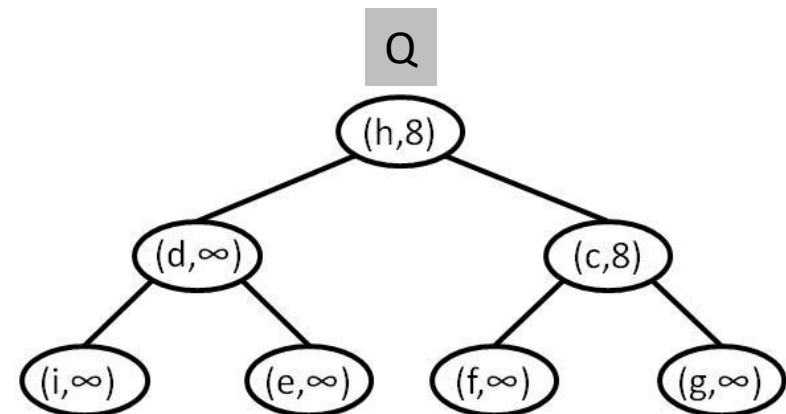


MST-PRIM(G, w, r)

```

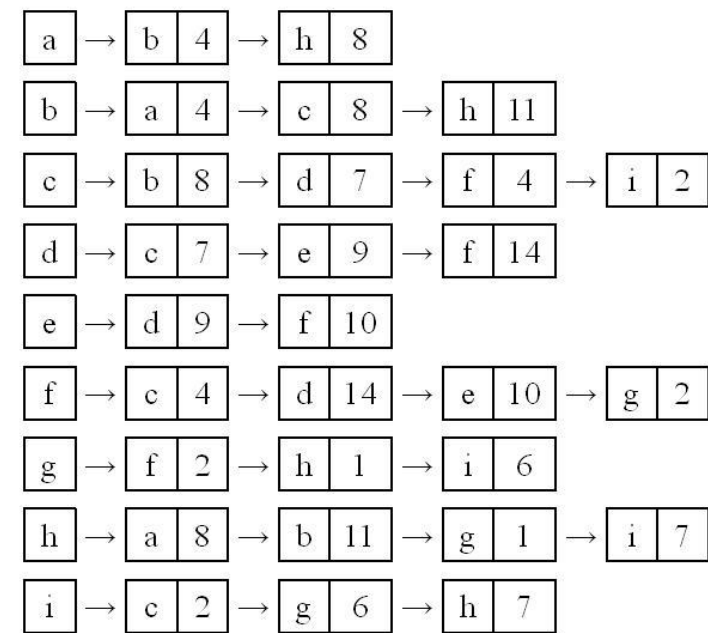
1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 

```



Contd...

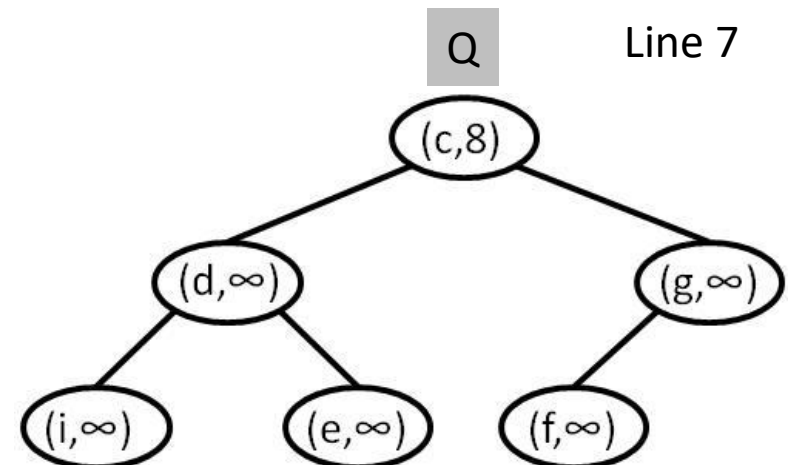
Vertex	π	key
a	NIL	0
b	a	4
c	b	8
d	NIL	∞
e	NIL	∞
f	NIL	∞
g	NIL	∞
h	a	8
i	NIL	∞



MST-PRIM(G, w, r)

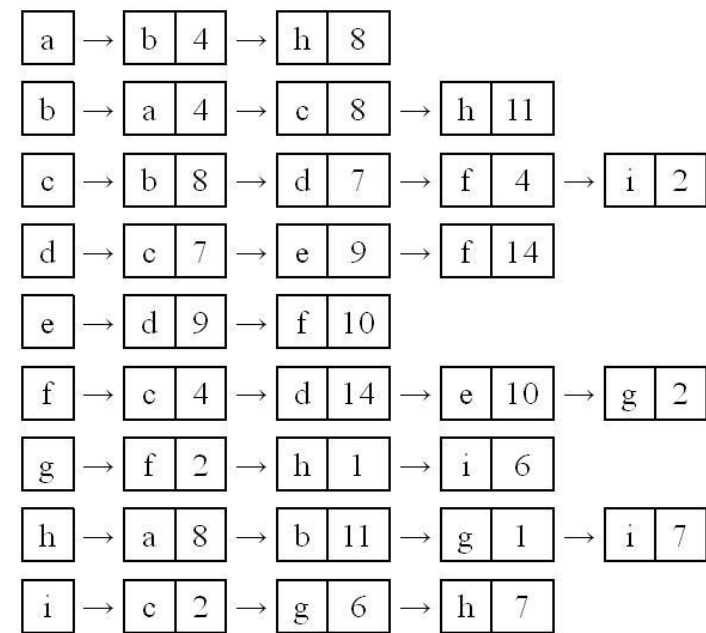
```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
  
```



Contd...

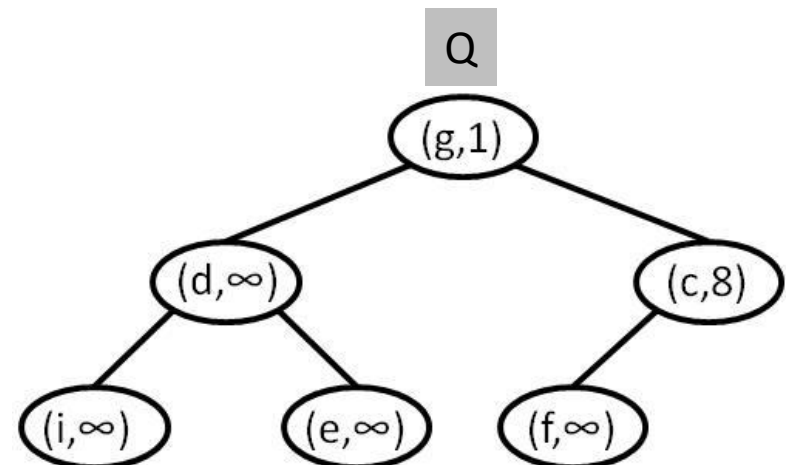
Vertex	π	key
a	NIL	0
b	a	4
c	b	8
d	NIL	∞
e	NIL	∞
f	NIL	∞
g	h	1
h	a	8
i	NIL	∞



MST-PRIM(G, w, r)

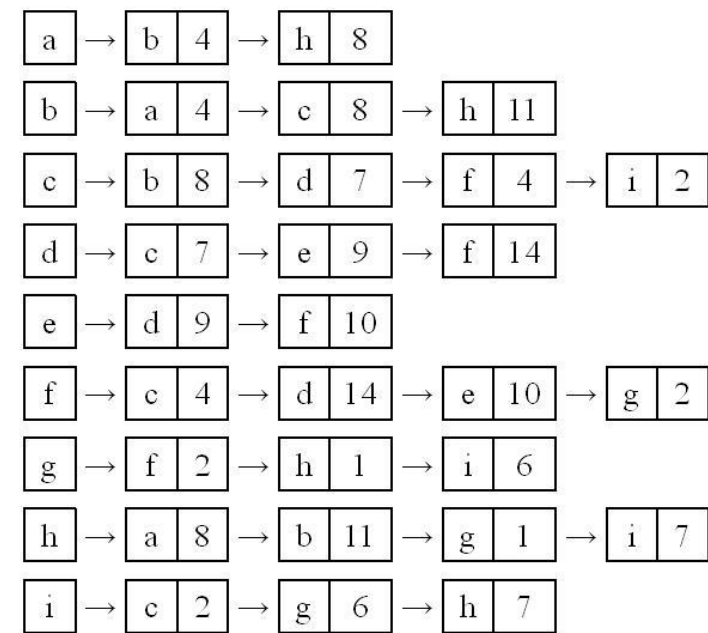
```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
  
```



Contd...

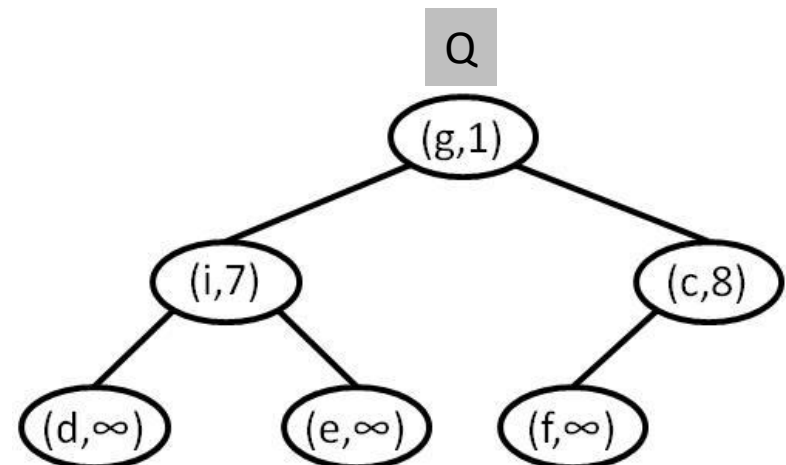
Vertex	π	key
a	NIL	0
b	a	4
c	b	8
d	NIL	∞
e	NIL	∞
f	NIL	∞
g	h	1
h	a	8
i	h	7



MST-PRIM(G, w, r)

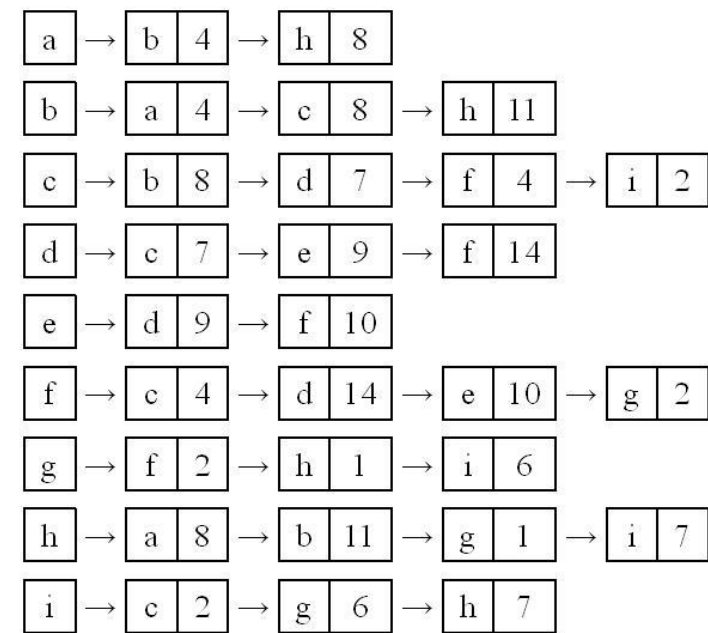
```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
  
```



Contd...

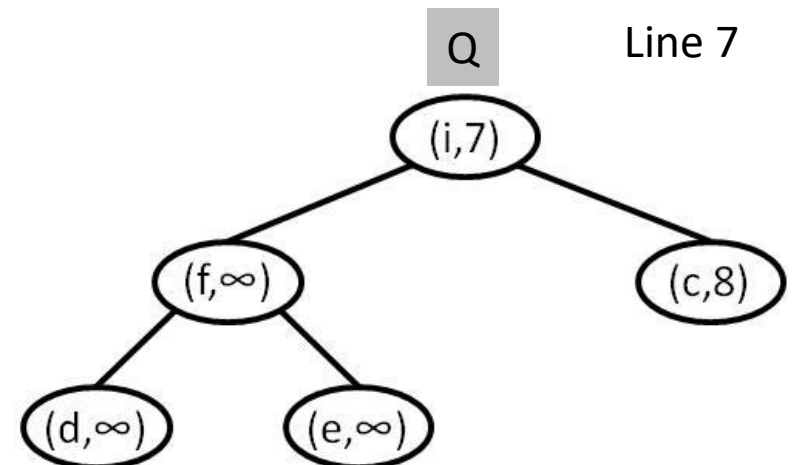
Vertex	π	key
a	NIL	0
b	a	4
c	b	8
d	NIL	∞
e	NIL	∞
f	NIL	∞
g	h	1
h	a	8
i	h	7



MST-PRIM(G, w, r)

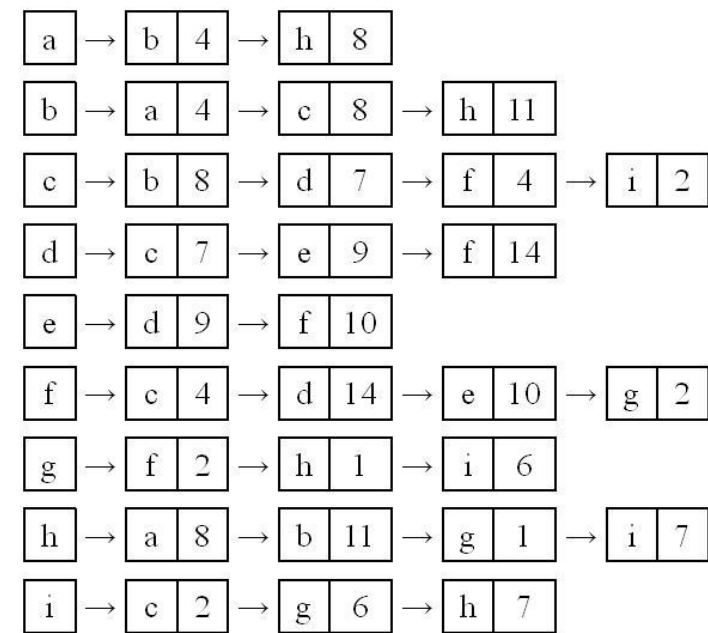
```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
  
```



Contd...

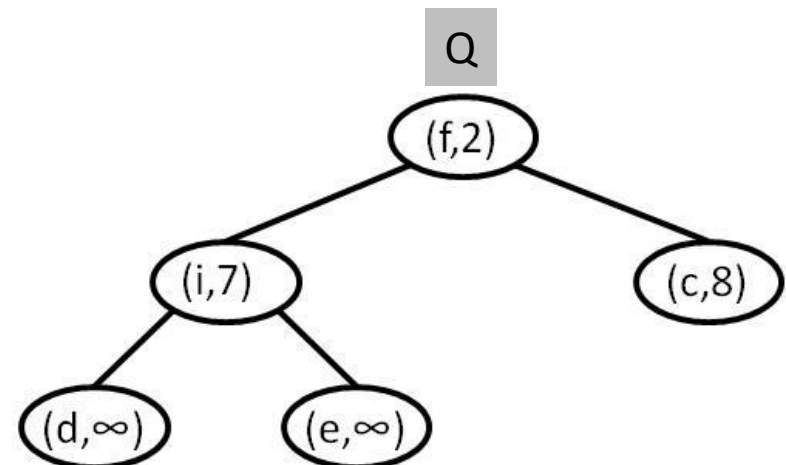
Vertex	π	key
a	NIL	0
b	a	4
c	b	8
d	NIL	∞
e	NIL	∞
f	g	2
g	h	1
h	a	8
i	h	7



MST-PRIM(G, w, r)

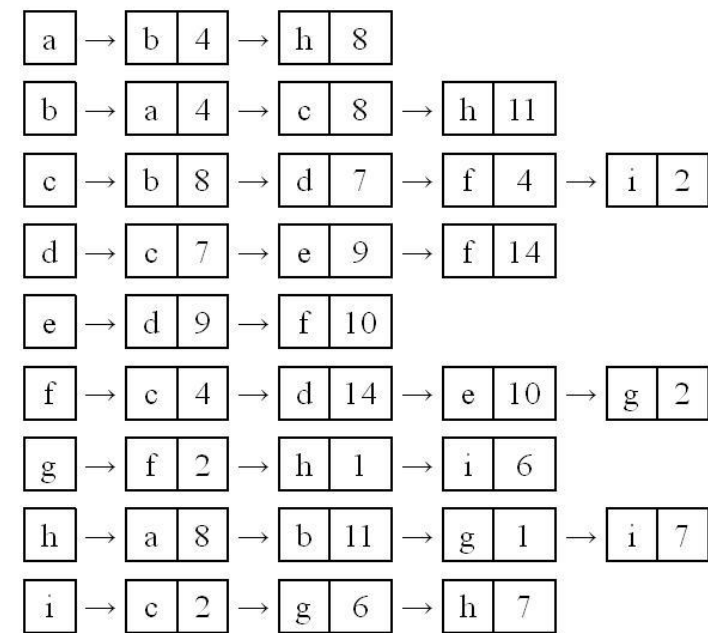
```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
  
```



Contd...

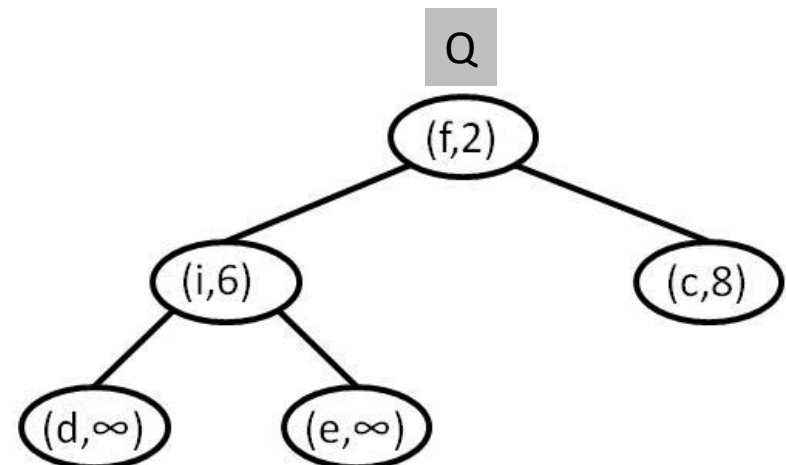
Vertex	π	key
a	NIL	0
b	a	4
c	b	8
d	NIL	∞
e	NIL	∞
f	g	2
g	h	1
h	a	8
i	g	6



MST-PRIM(G, w, r)

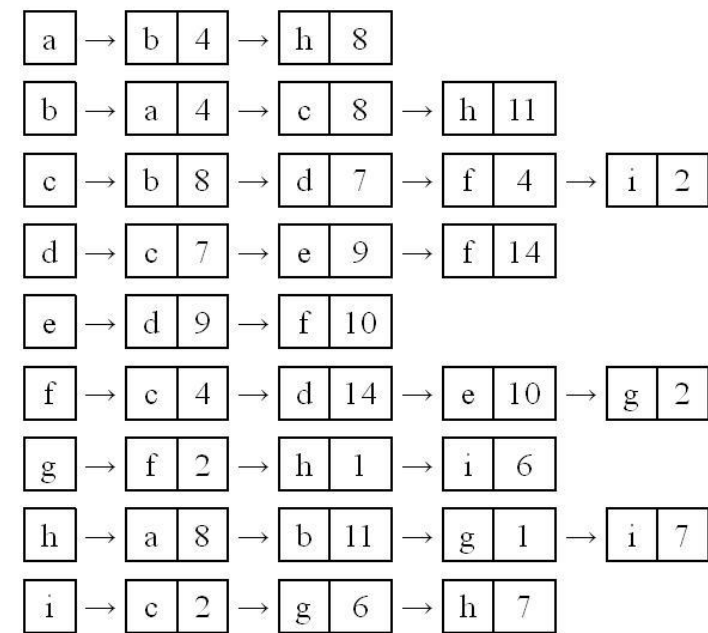
```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
  
```



Contd...

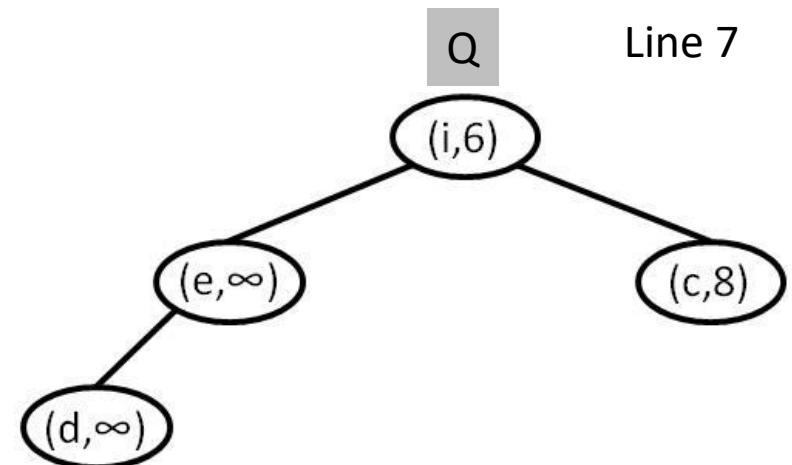
Vertex	π	key
a	NIL	0
b	a	4
c	b	8
d	NIL	∞
e	NIL	∞
f	g	2
g	h	1
h	a	8
i	g	6



MST-PRIM(G, w, r)

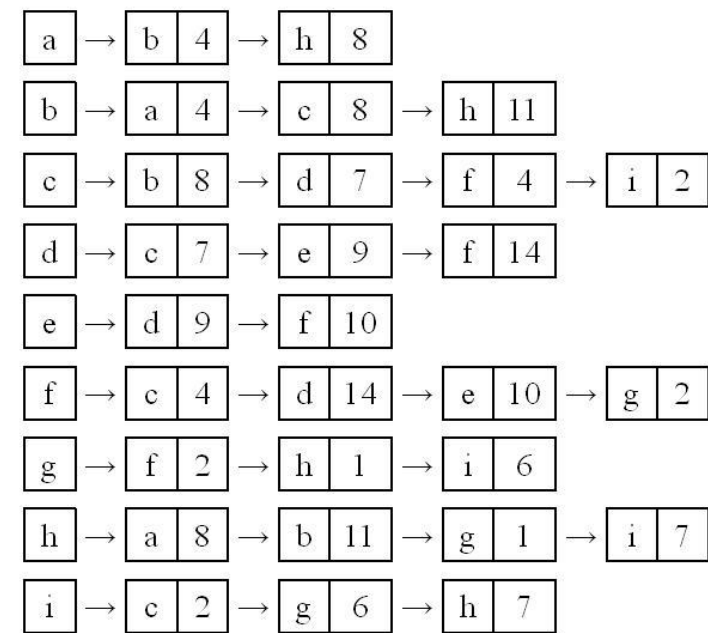
```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
  
```



Contd...

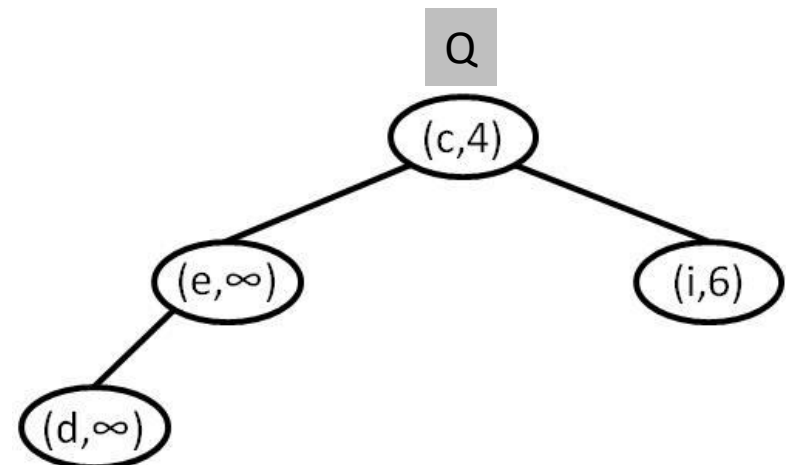
Vertex	π	key
a	NIL	0
b	a	4
c	f	4
d	NIL	∞
e	NIL	∞
f	g	2
g	h	1
h	a	8
i	g	6



MST-PRIM(G, w, r)

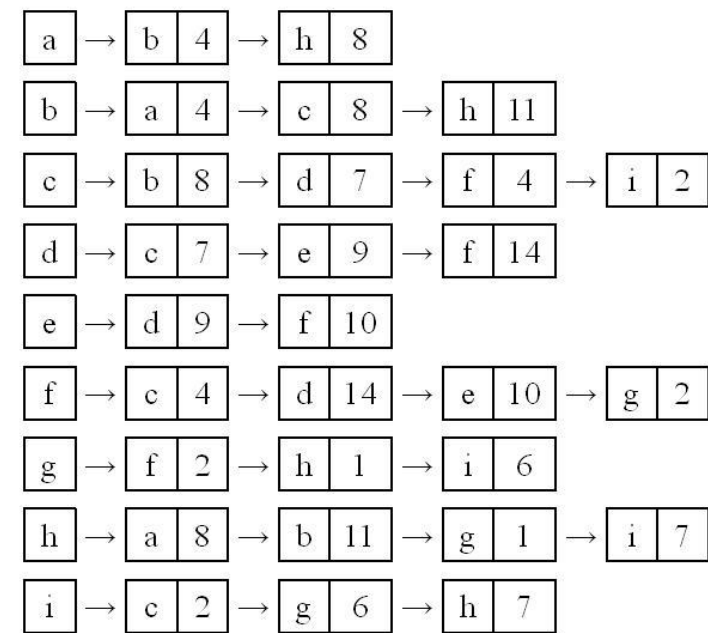
```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
  
```



Contd...

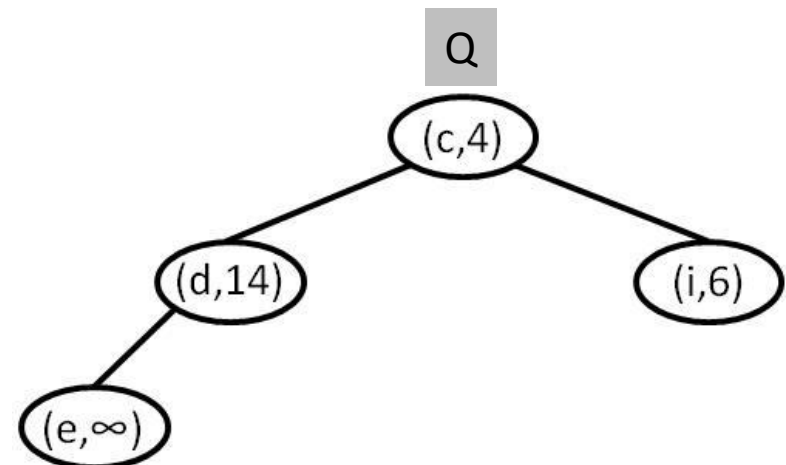
Vertex	π	key
a	NIL	0
b	a	4
c	f	4
d	f	14
e	NIL	∞
f	g	2
g	h	1
h	a	8
i	g	6



MST-PRIM(G, w, r)

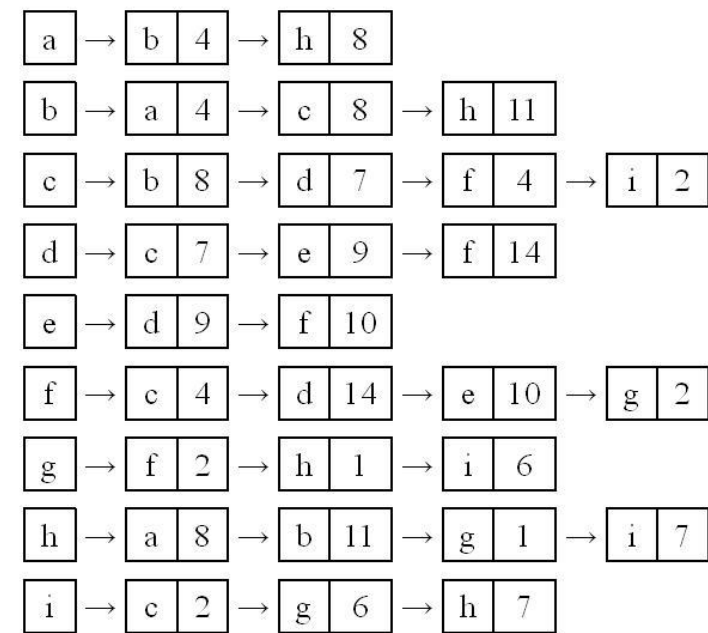
```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
  
```



Contd...

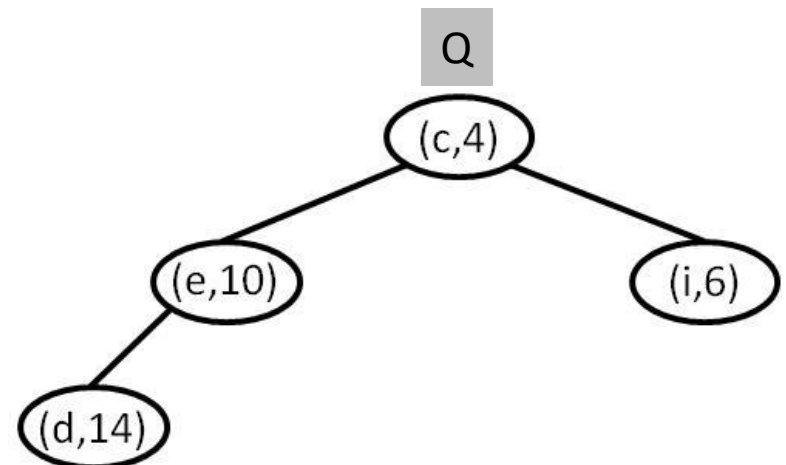
Vertex	π	key
a	NIL	0
b	a	4
c	f	4
d	f	14
e	f	10
f	g	2
g	h	1
h	a	8
i	g	6



MST-PRIM(G, w, r)

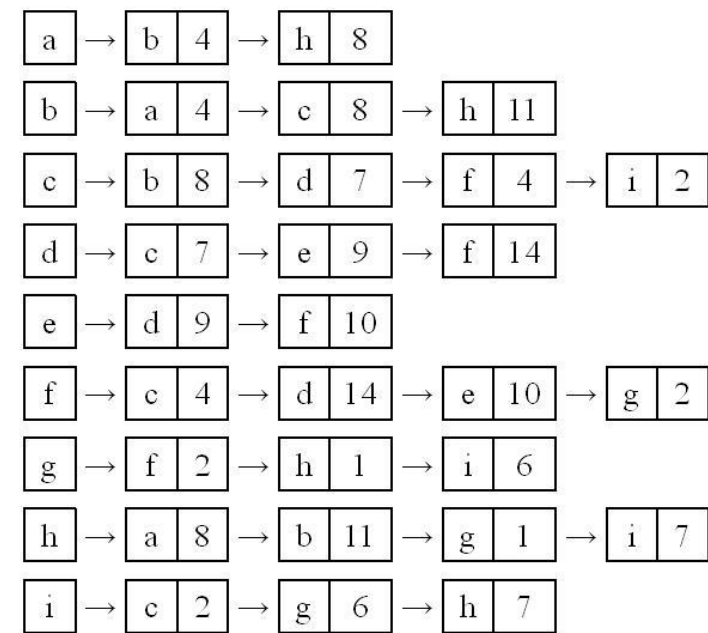
```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
  
```



Contd...

Vertex	π	key
a	NIL	0
b	a	4
c	f	4
d	f	14
e	f	10
f	g	2
g	h	1
h	a	8
i	g	6

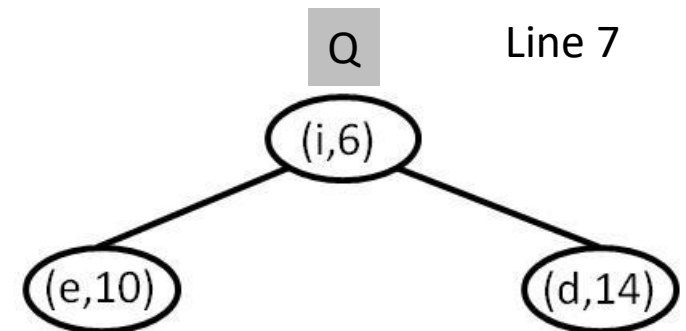


MST-PRIM(G, w, r)

```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 

```



Contd...

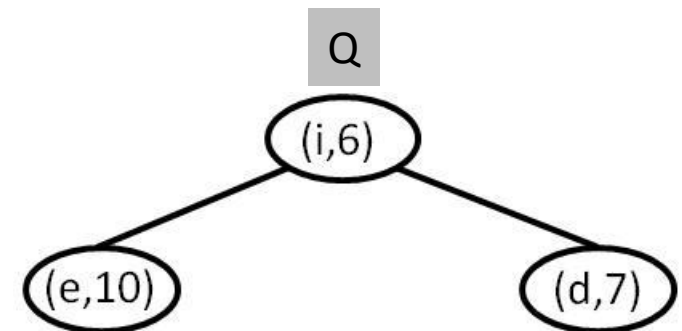
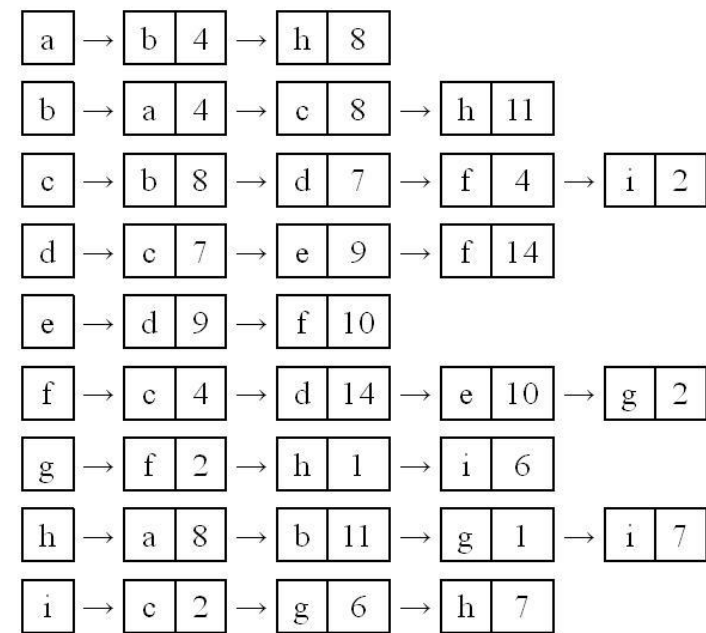
Vertex	π	key
a	NIL	0
b	a	4
c	f	4
d	c	7
e	f	10
f	g	2
g	h	1
h	a	8
i	g	6

MST-PRIM(G, w, r)

```

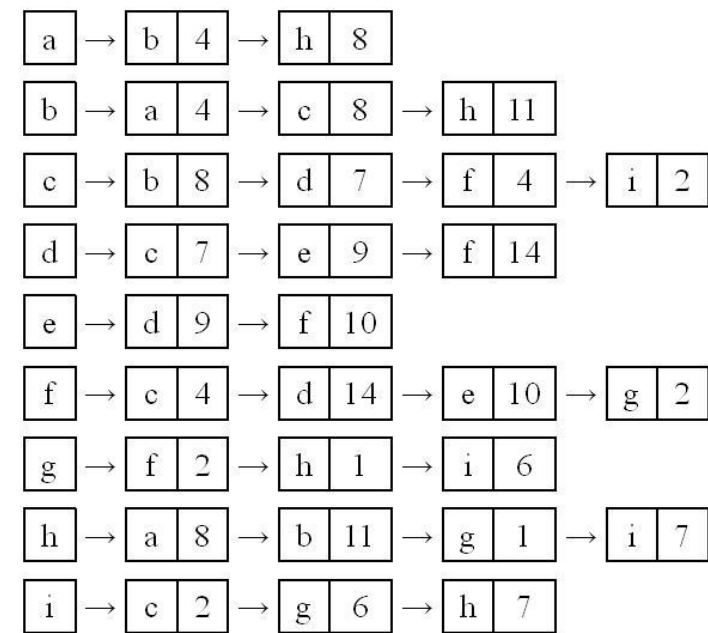
1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 

```



Contd...

Vertex	π	key
a	NIL	0
b	a	4
c	f	4
d	c	7
e	f	10
f	g	2
g	h	1
h	a	8
i	c	2

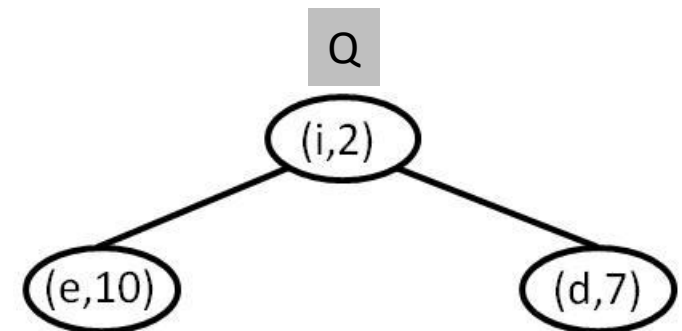


MST-PRIM(G, w, r)

```

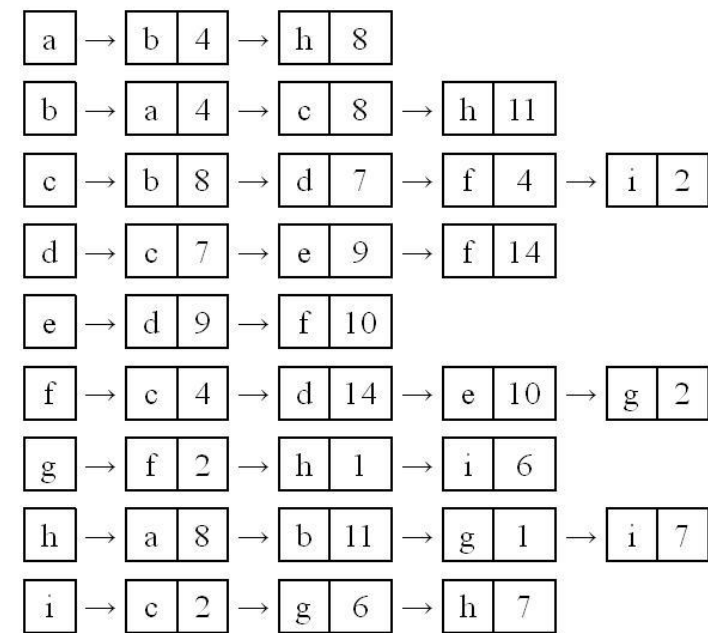
1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 

```



Contd...

Vertex	π	key
a	NIL	0
b	a	4
c	f	4
d	c	7
e	f	10
f	g	2
g	h	1
h	a	8
i	c	2

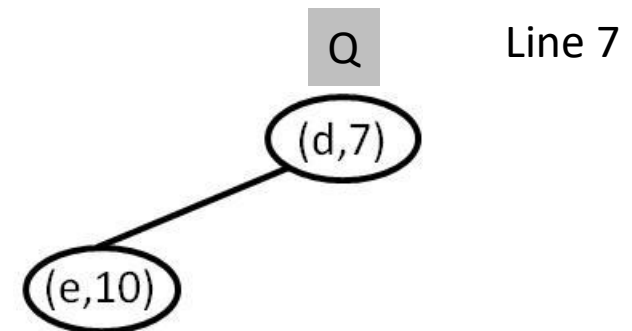


MST-PRIM(G, w, r)

```

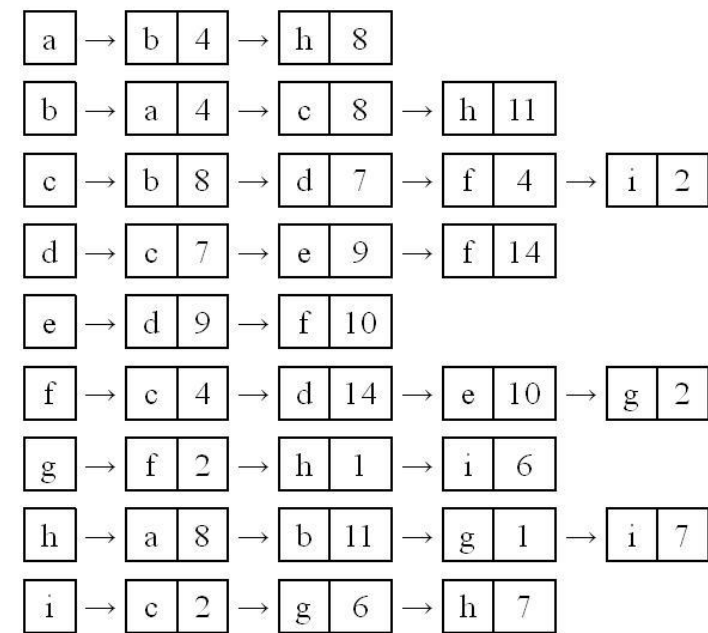
1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 

```



Contd...

Vertex	π	key
a	NIL	0
b	a	4
c	f	4
d	c	7
e	f	10
f	g	2
g	h	1
h	a	8
i	c	2



Q

Line 7

(e,10)

MST-PRIM(G, w, r)

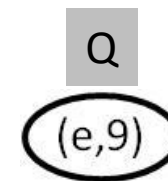
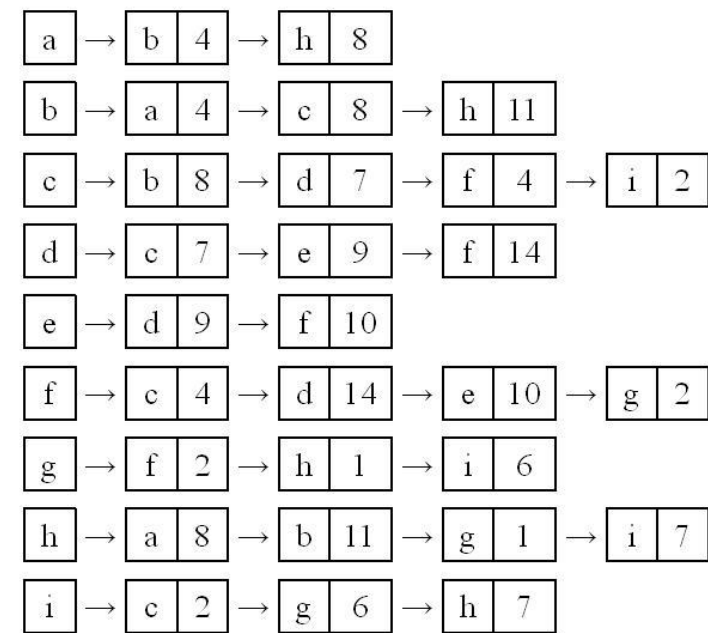
```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 

```

Contd...

Vertex	π	key
a	NIL	0
b	a	4
c	f	4
d	c	7
e	d	9
f	g	2
g	h	1
h	a	8
i	c	2



MST-PRIM(G, w, r)

```

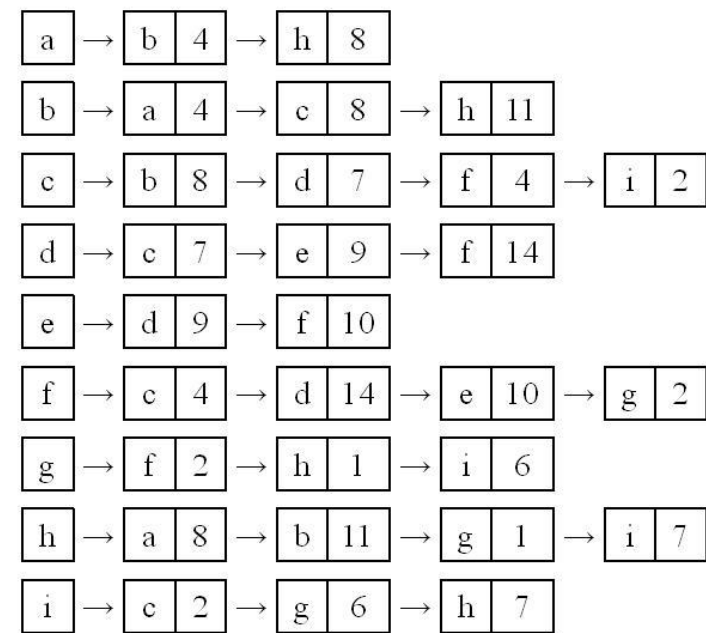
1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 

```

Contd...

$A = \{(b,a), (c,f), (d,c), (e,d), (f,g), (g,h), (h,a), (i,c)\}$

Vertex	π	key
a	NIL	0
b	a	4
c	f	4
d	c	7
e	d	9
f	g	2
g	h	1
h	a	8
i	c	2



Q

Line 7

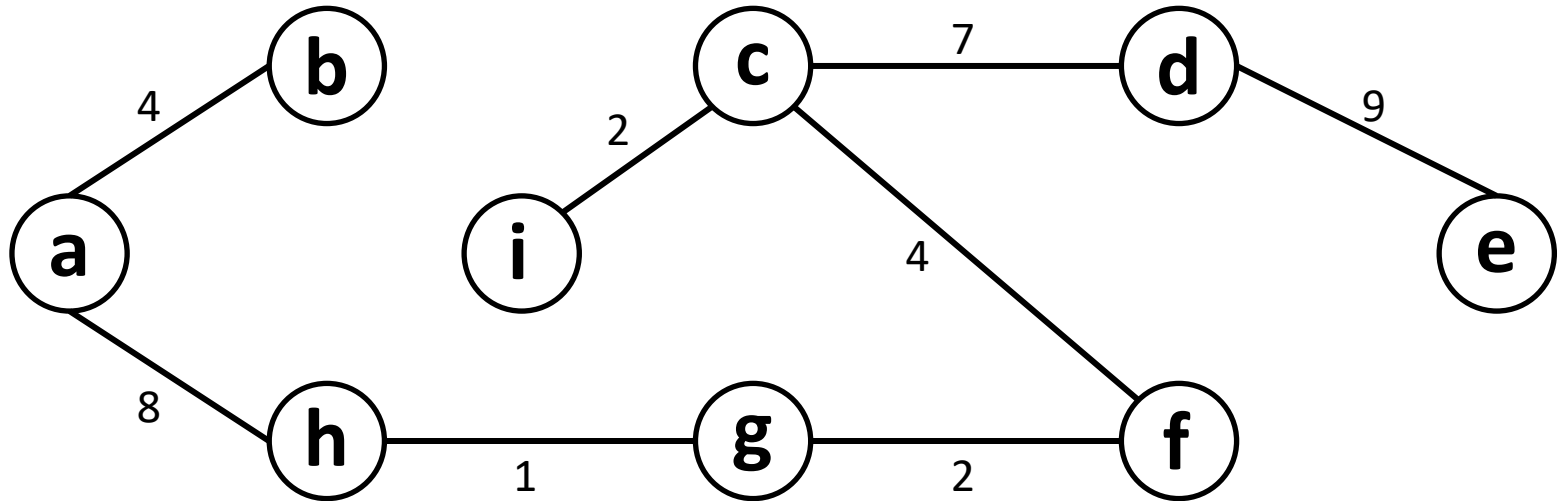
MST-PRIM(G, w, r)

```

1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
  
```


Contd...

$A = \{(b,a), (c,f), (d,c), (e,d), (f,g), (g,h), (h,a), (i,c)\}$



Cost = 37