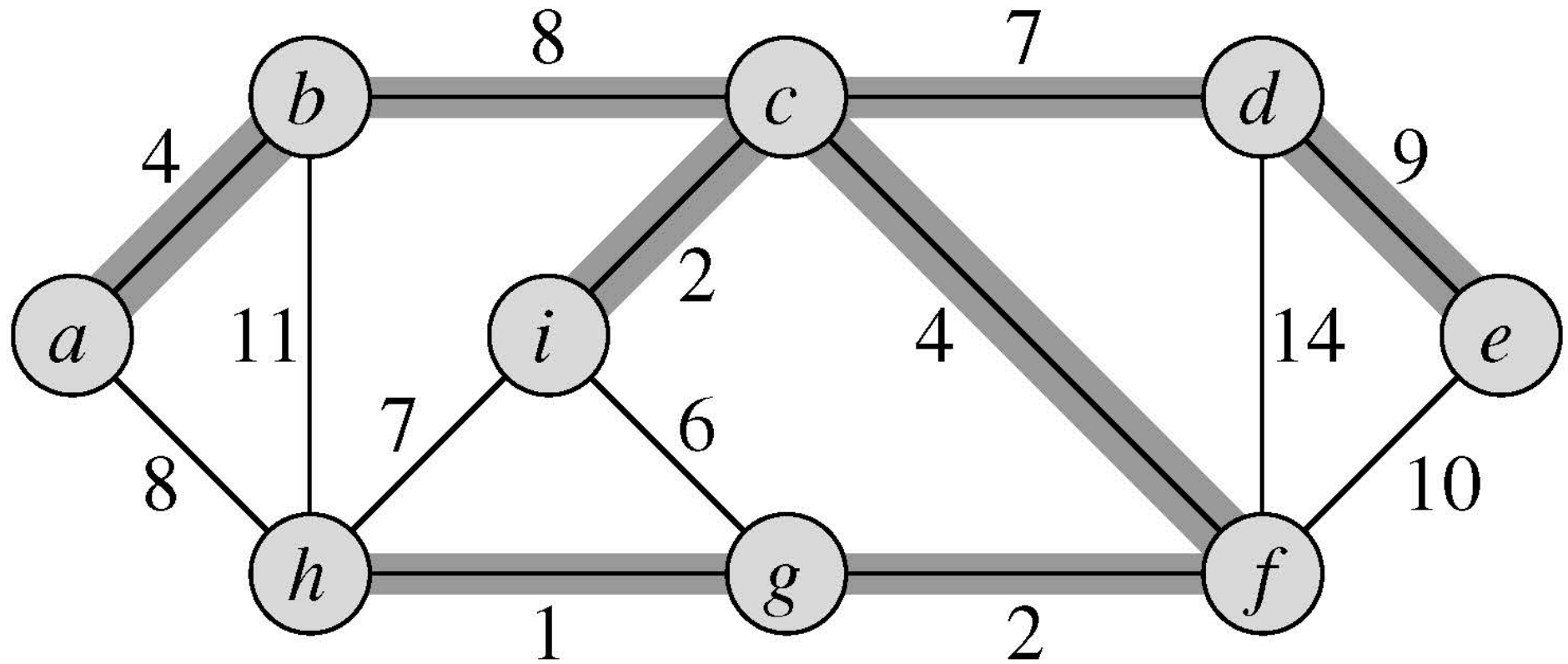


Minimale aufspannende Bäume

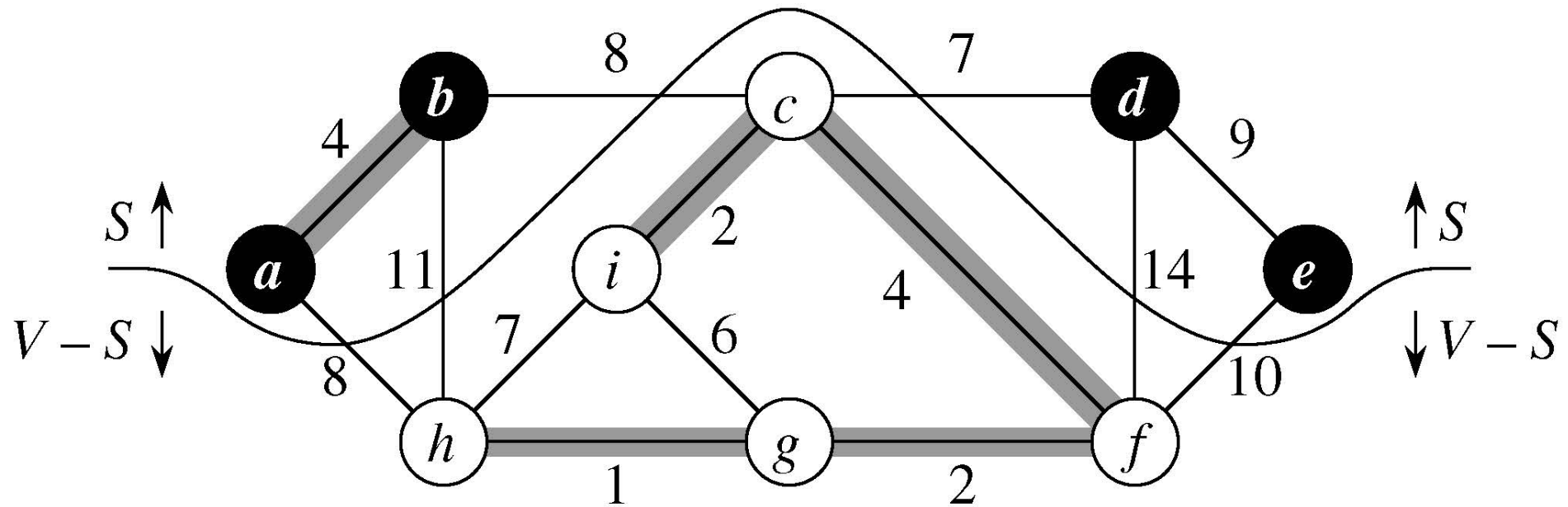
Beispiel



GENERIC-MST(G, w)


```
1   $A \leftarrow \emptyset$   
2  while  $A$  does not form a spanning tree  
3      do find an edge  $(u, v)$  that is safe for  $A$   
4       $A \leftarrow A \cup \{(u, v)\}$   
5  return  $A$ 
```

Schnitt



Kruskal-Algorithmus

MST-KRUSKAL(G, w)

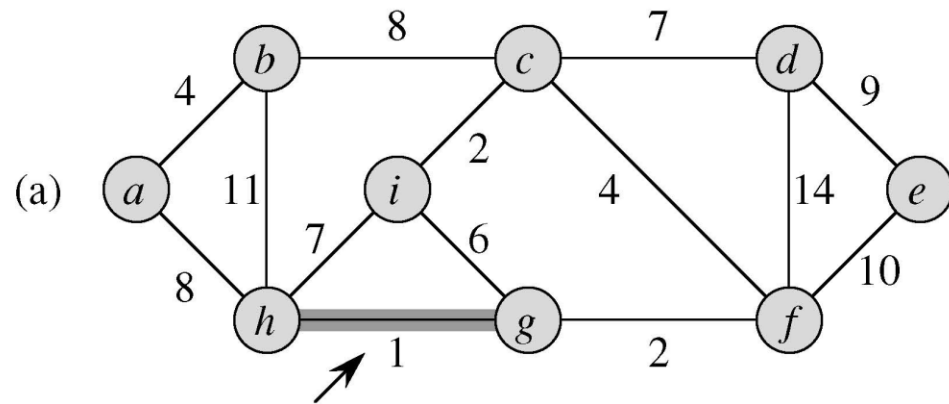
```
1   $A \leftarrow \emptyset$ 
2  for each vertex  $v \in V[G]$ 
3      do MAKE-SET( $v$ )
4  sort the edges of  $E$  into nondecreasing order by weight  $w$ 
5  for each edge  $(u, v) \in E$ , taken in nondecreasing order by weight
6      do if 
7          then  $A \leftarrow A \cup \{(u, v)\}$ 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

Kruskal-Algorithmus

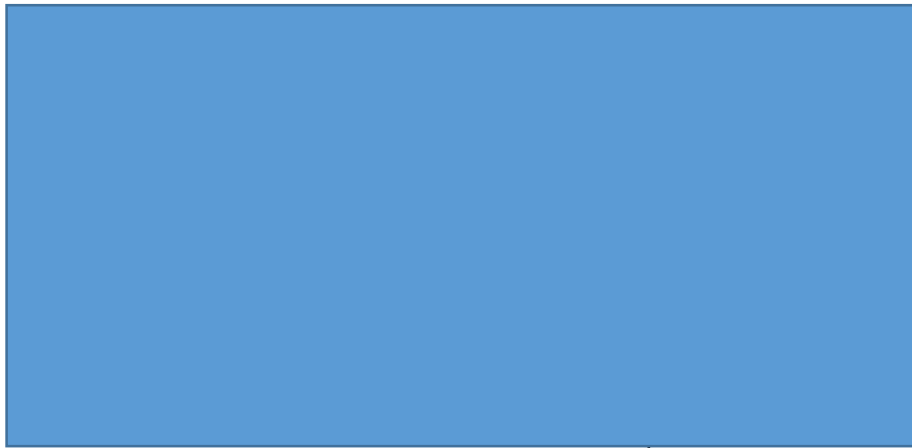
MST-KRUSKAL(G, w)

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

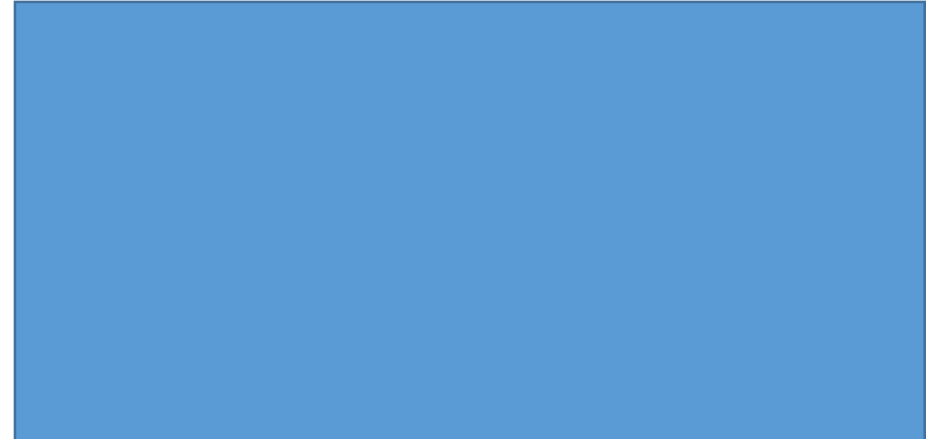
Kruskal Beispiel I



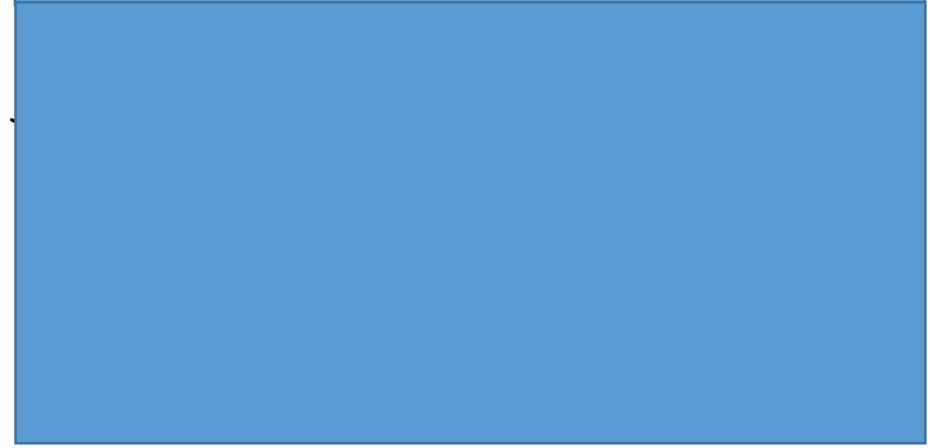
(c)



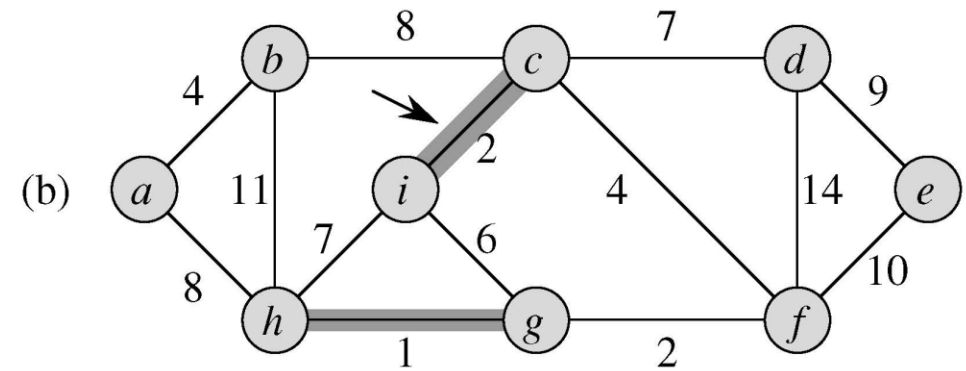
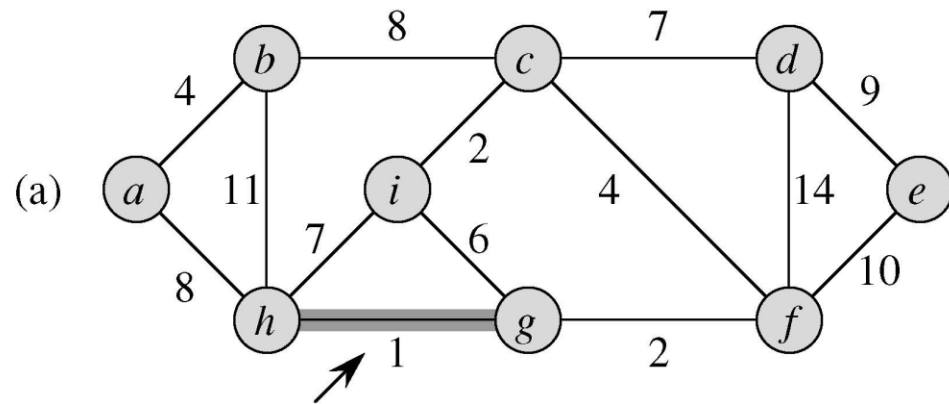
(b)



(d)



Kruskal Beispiel I



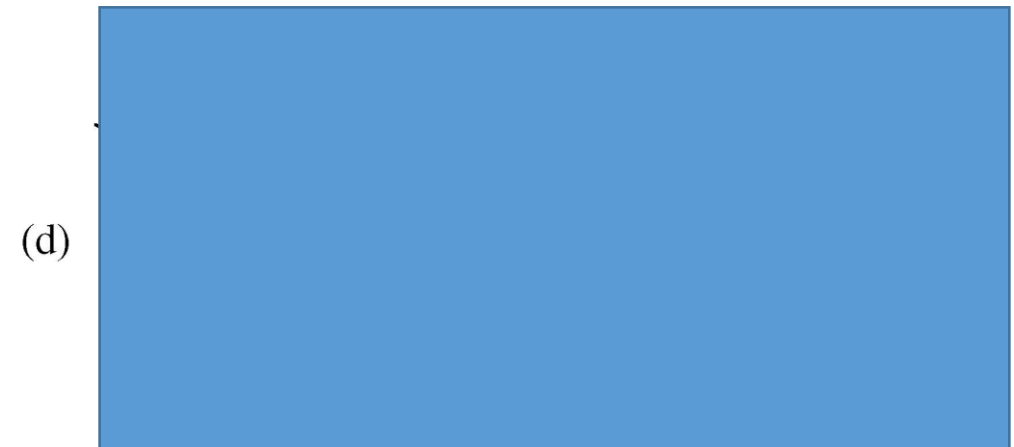
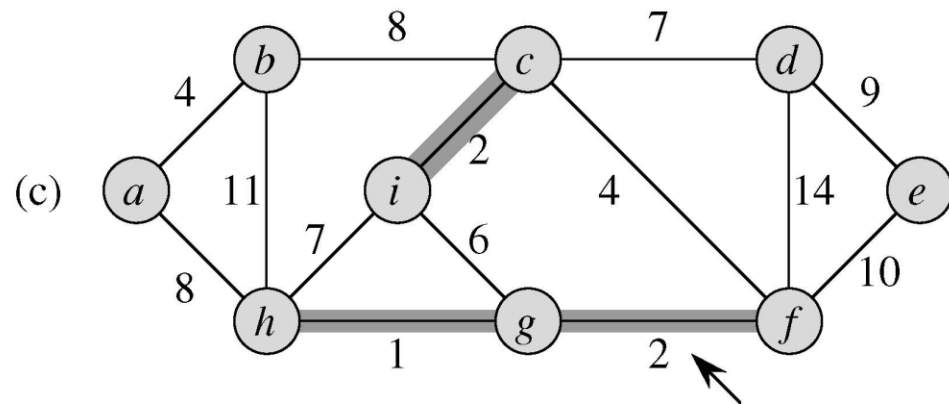
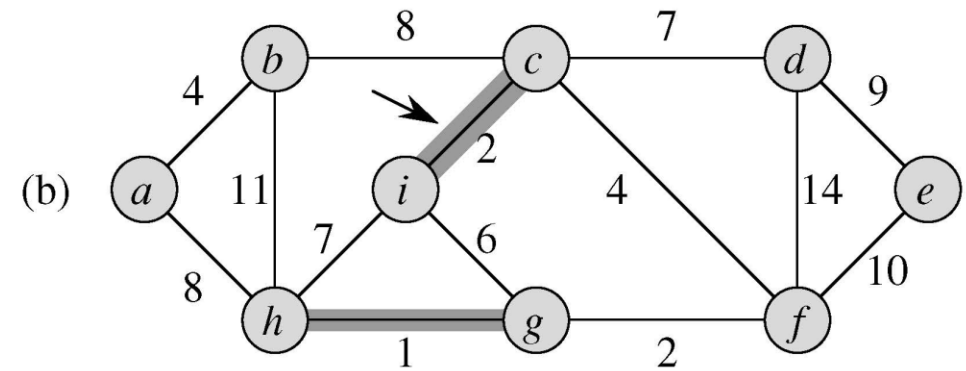
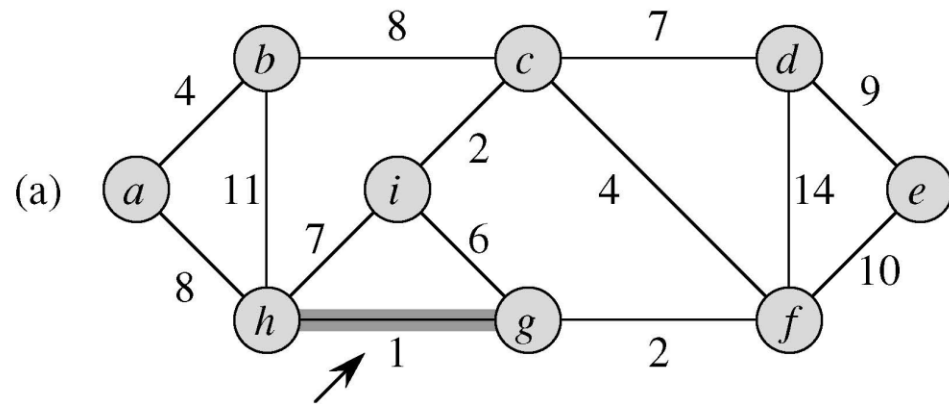
(c)



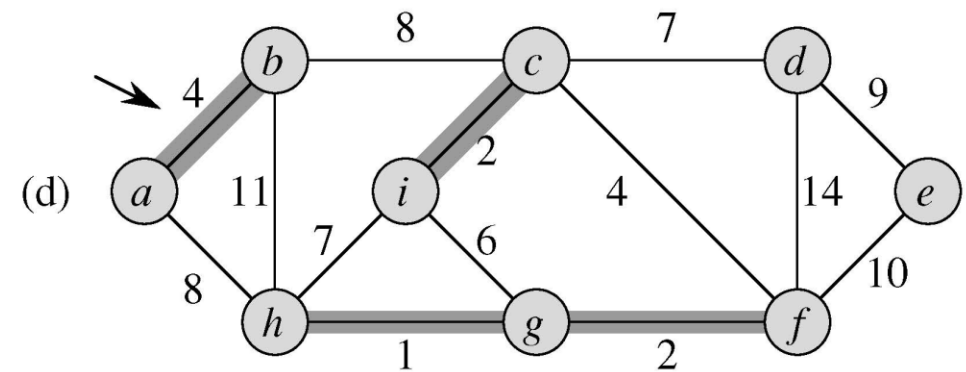
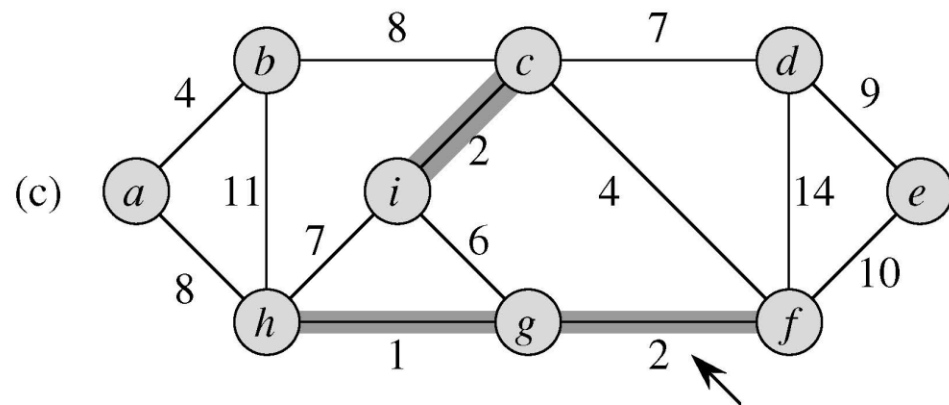
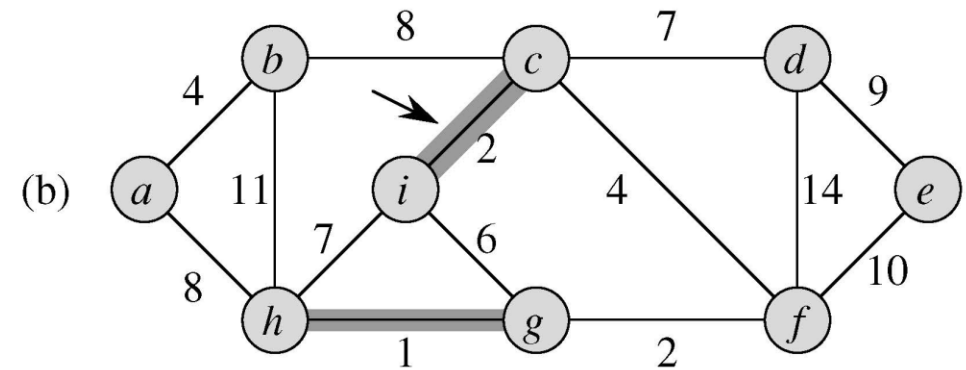
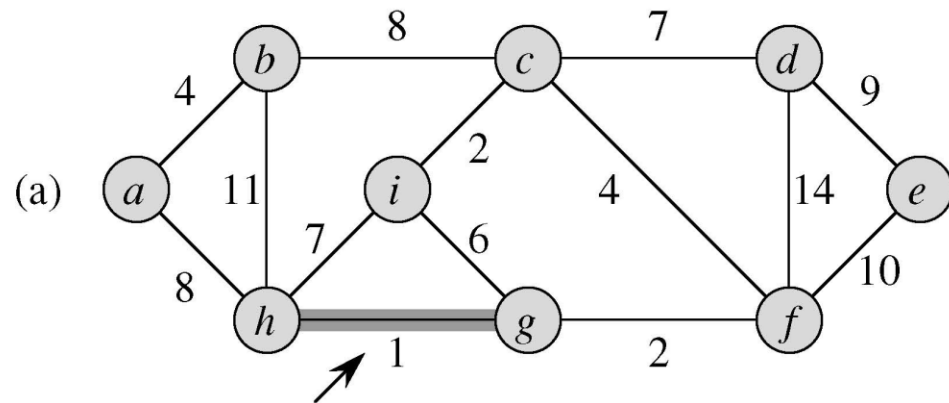
(d)



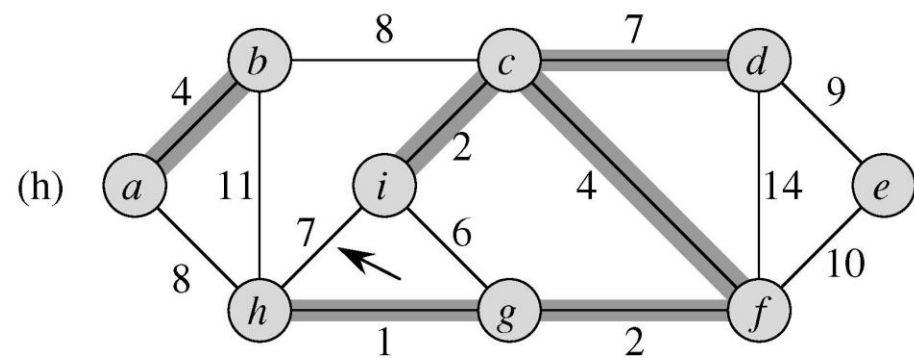
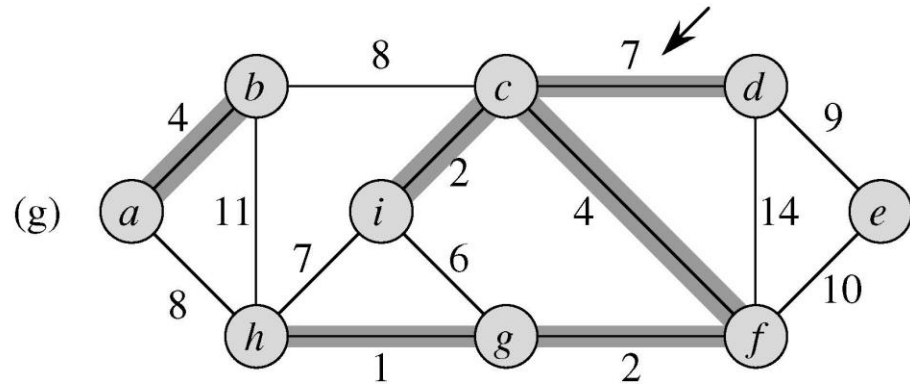
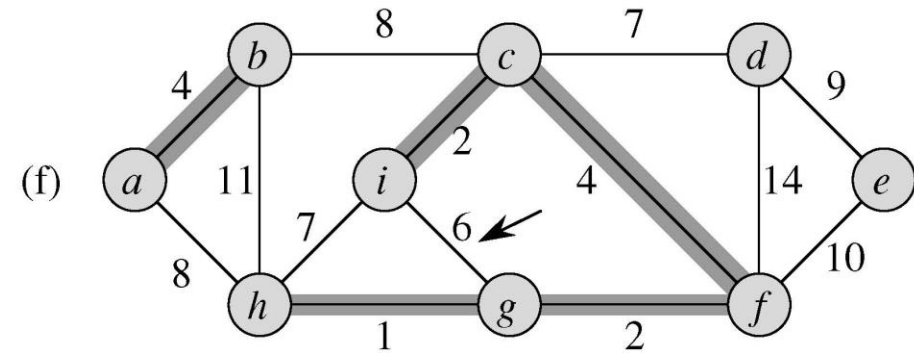
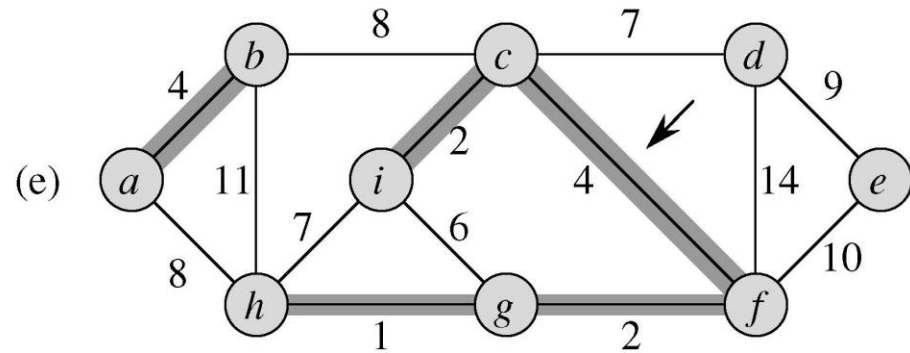
Kruskal Beispiel I



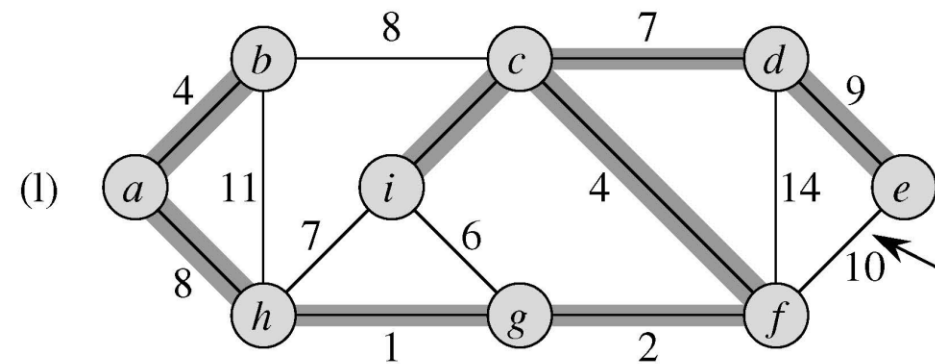
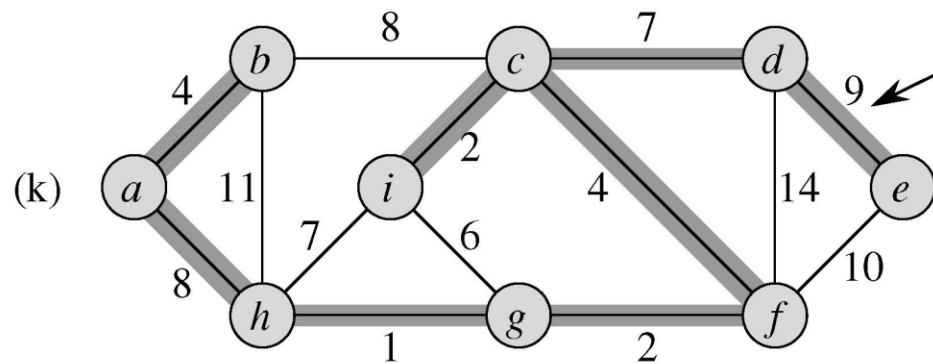
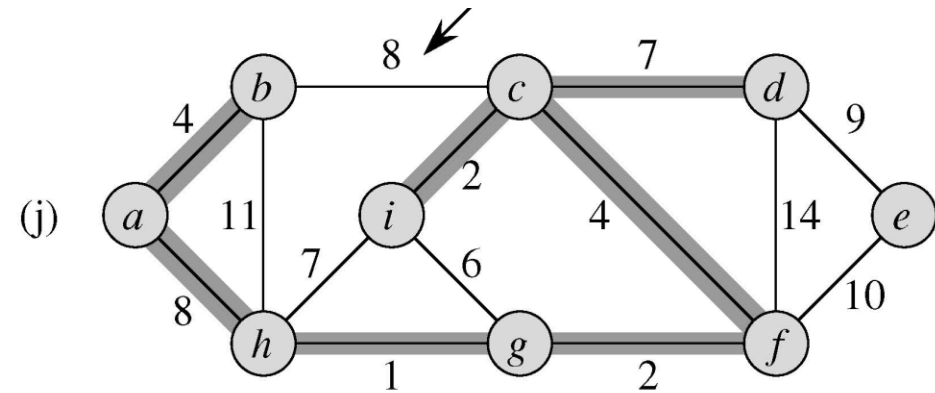
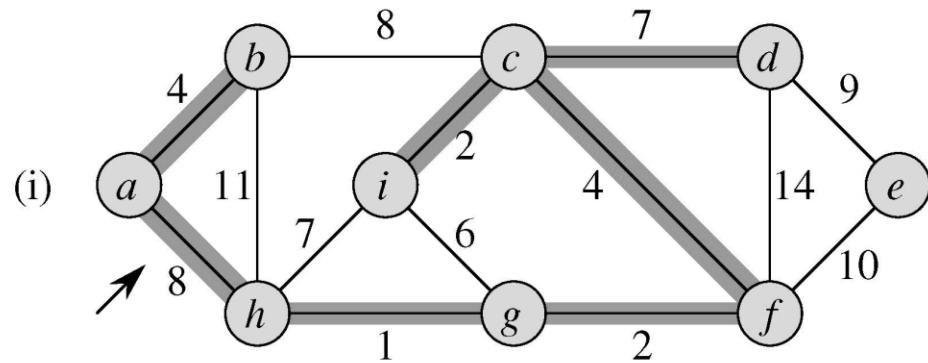
Kruskal Beispiel I



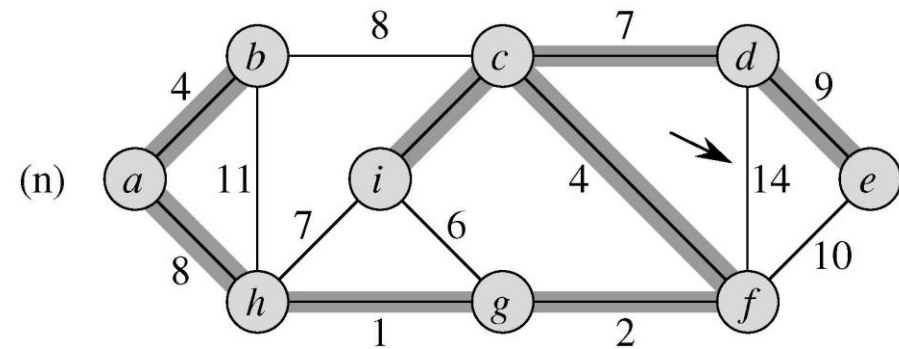
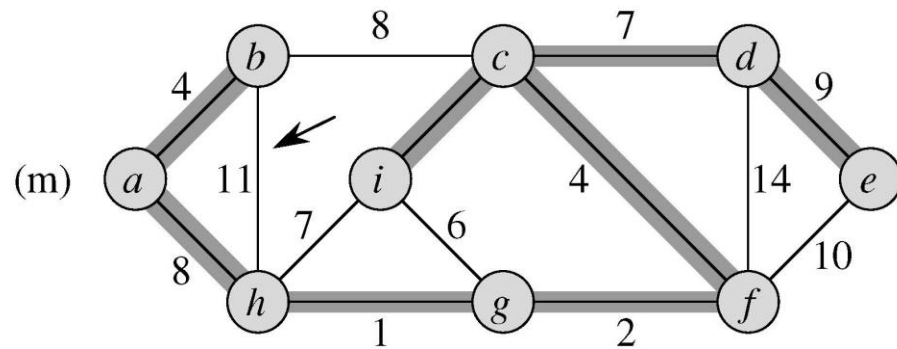
Kruskal Beispiel II



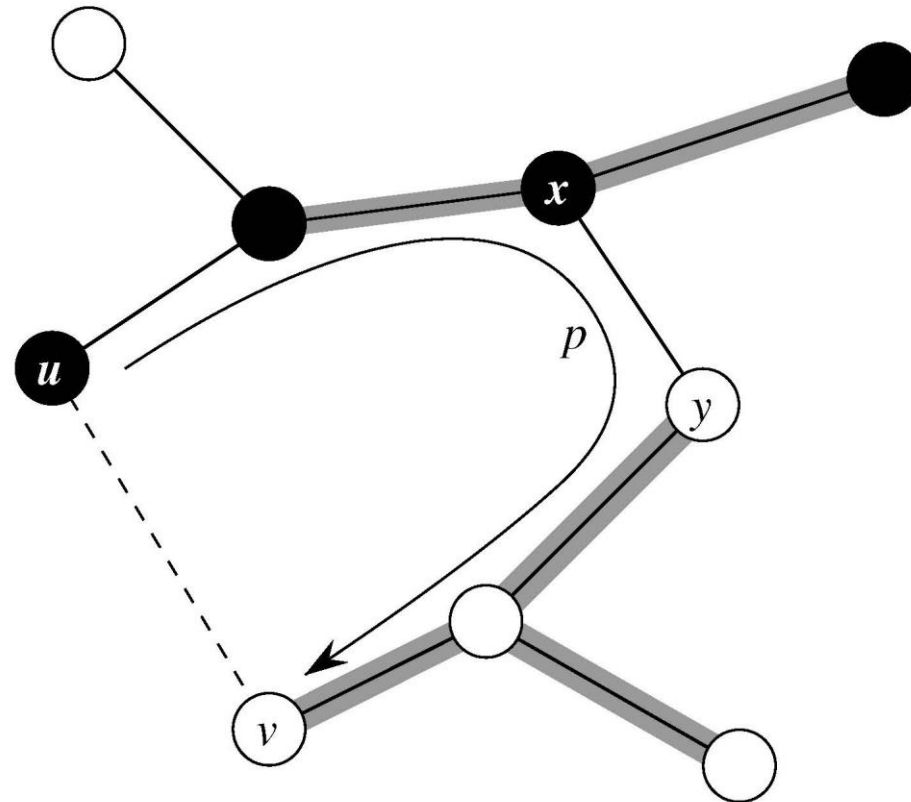
Kruskal Beispiel III



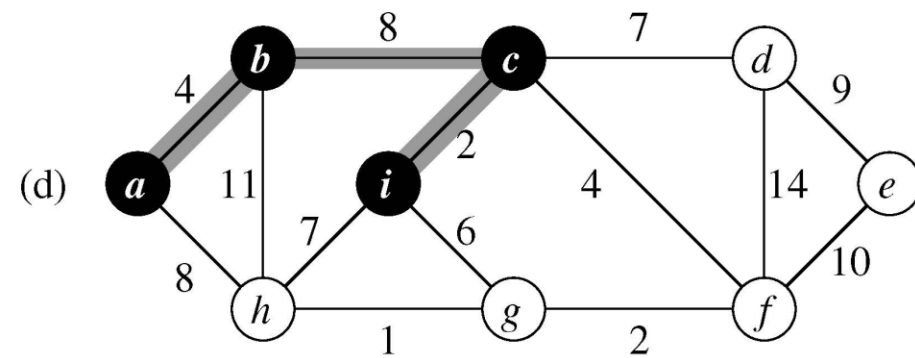
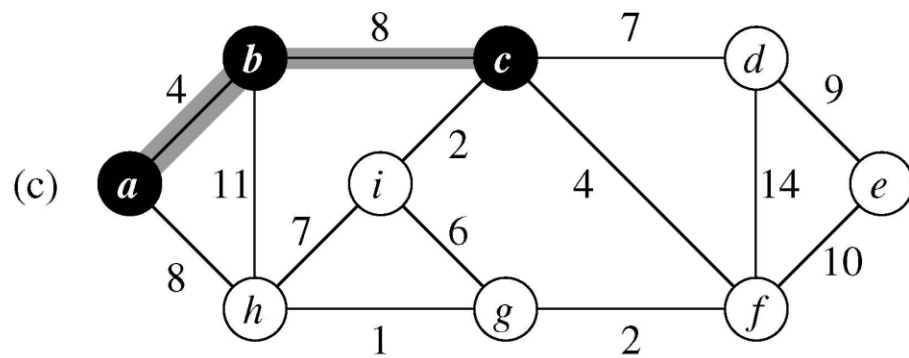
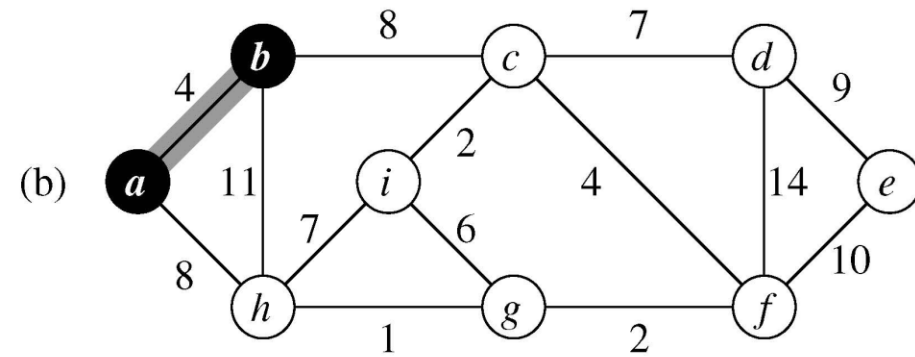
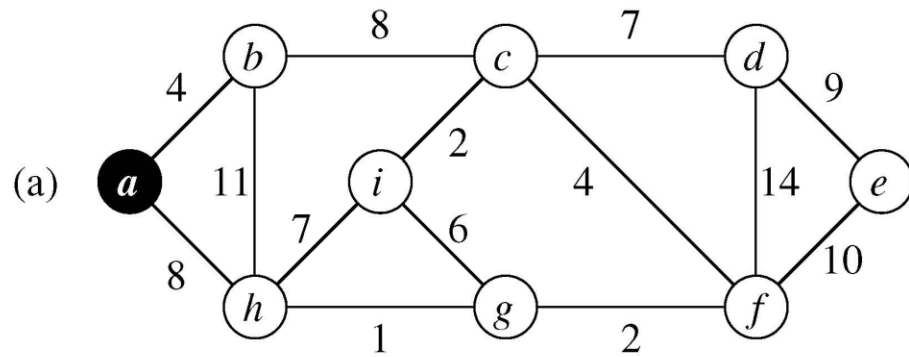
Kruskal Beispiel IV



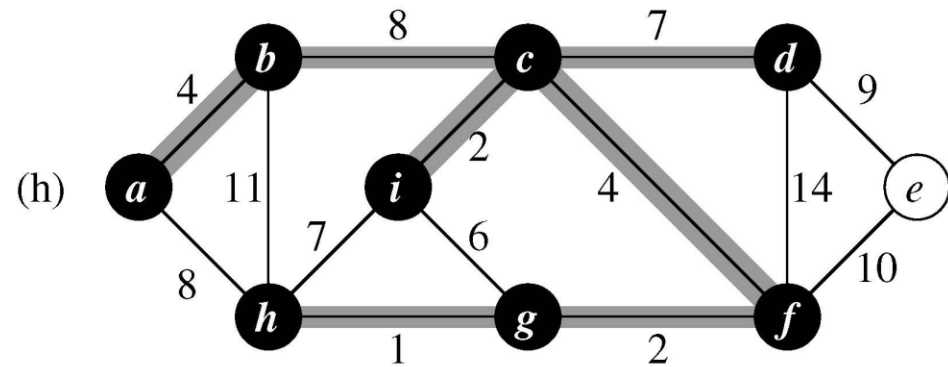
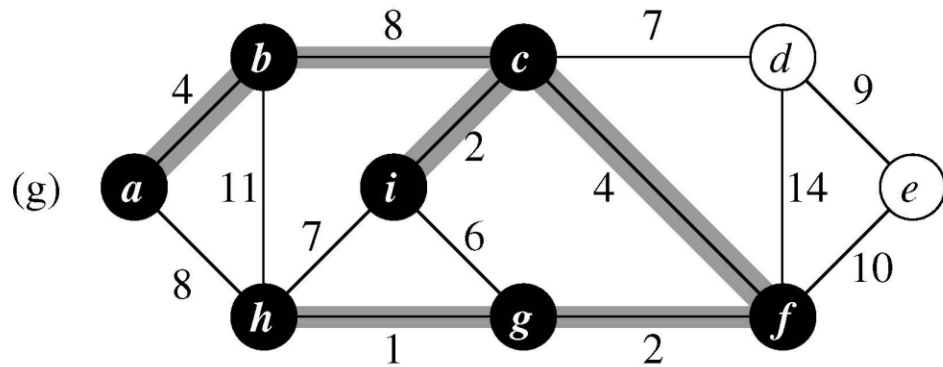
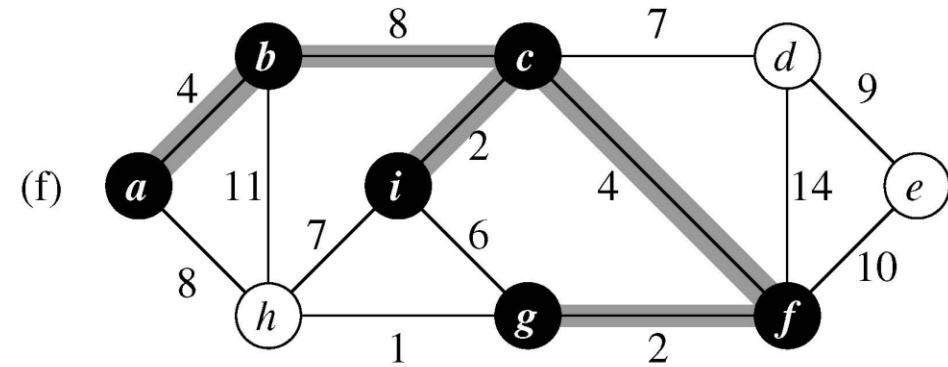
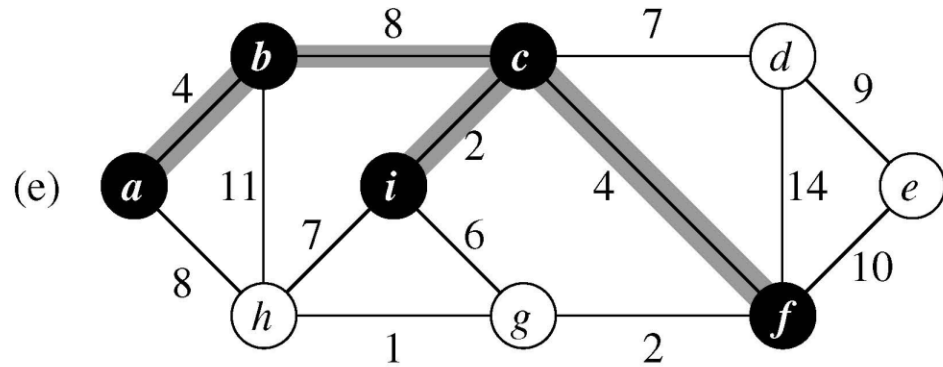
Zum Kruskal-Korrektheits-Beweis



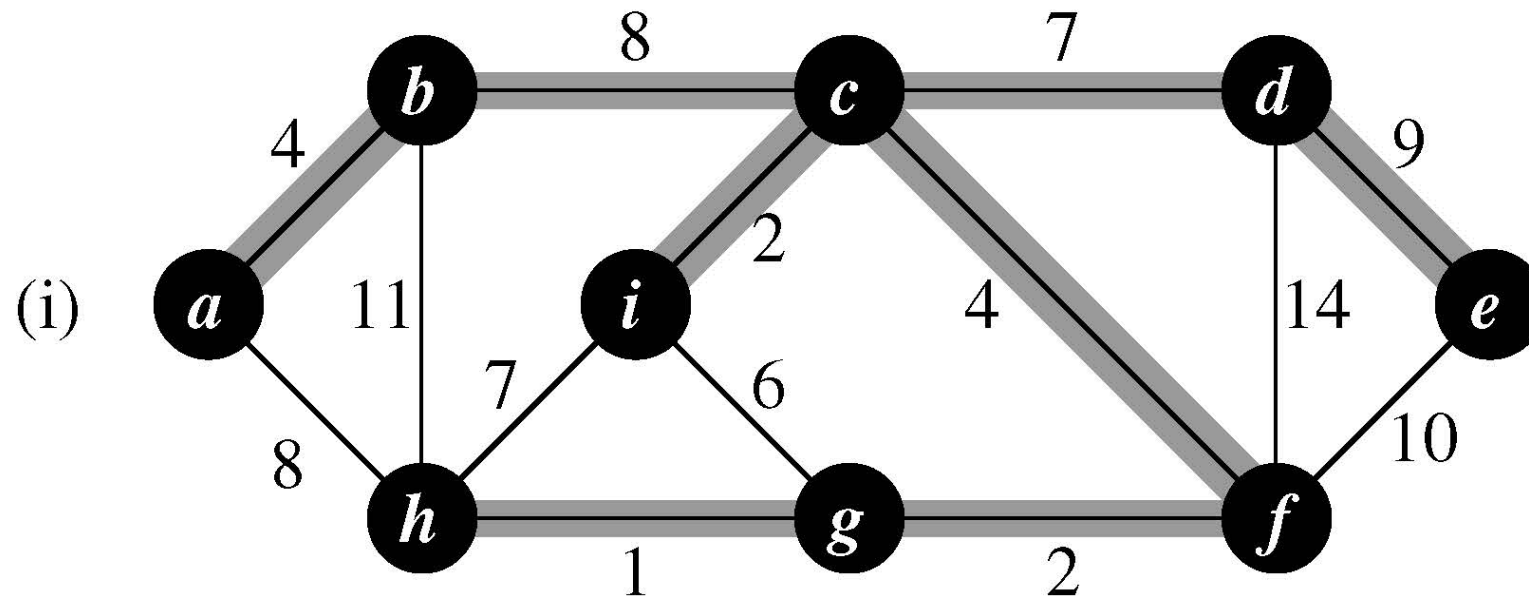
Beispiel Prim



Beispiel Prim II




Beispiel Prim III



Algorithmus von Prim

MST-PRIM(G, w, r)

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

Algorithmus von Prim

MST-PRIM(G, w, r)

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