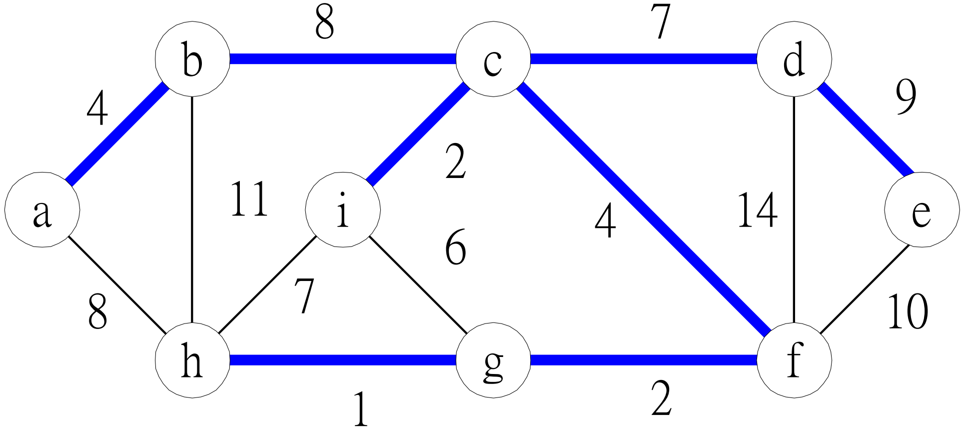
Ch.23. Minimum spanning tree

Read the entire chapter, but omit proof of Theorem 23.1

Let G=(V,E) be a connected, undirected graph. For each edge $(u,v) \in E$, we have a weight $w(u,v)$ specifying the cost to connect u and v. We wish to find an acyclic subset $T \subseteq E$ that connects all of the vertices and whose total weight $w(T) = \sum_{(u,v)\in T} w(u,v)$ is minimized. Since T is acyclic and connects all of the vertices, it must form a tree, which we call a ***spanning tree***. We call the problem of determine the tree T the ***minimum spanning tree problem***.



23.1 Growing a minimum spanning tree

GENERIC-MST(*G*, *w*)

$A = \phi$

while *A* does not form a spanning tree

    find an edge (*u*, *v*) that is ***safe*** for *A*

$A = A \cup \{(u, v)\}$

return *A*

What is a safe edge?

1. One that has the smallest cost among edges not yet in the MST
2. One that won't create a cycle

This is an example of a **Greedy Algorithm**:
algorithms that solve a problem in stages, with the best possible partial solution computed or chosen at each stage.

23.2 The algorithms of Kruskal and Prim

Kruskal's Algorithm

Uses the Disjoint Set data structure to store nodes and Find & Union operations

Prim's Algorithm

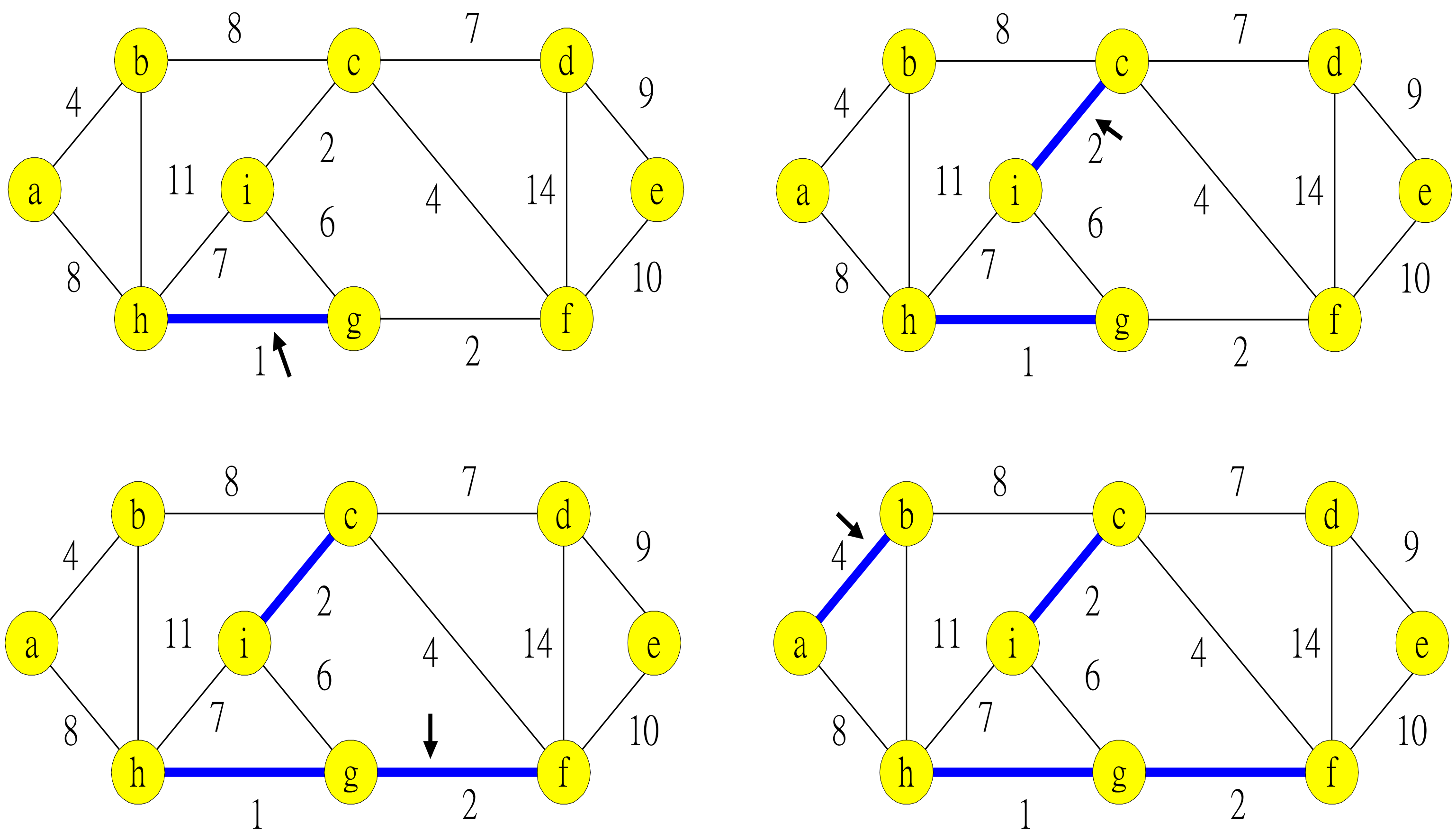
Uses a Min-Priority-Queue and Extract-Min operation

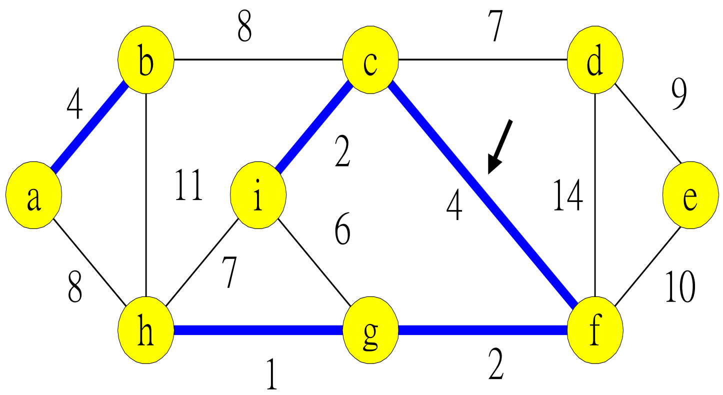
23.2 The algorithm of Kruskal

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

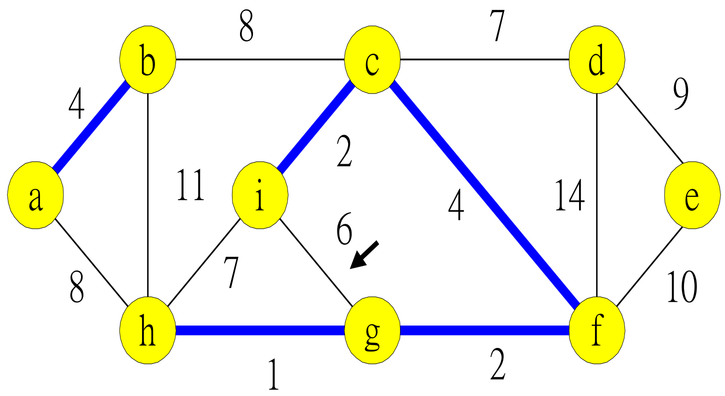
Complexity $O(E \log V)$

Reading Assignment: Read and understand this complexity calculation from text p. 633

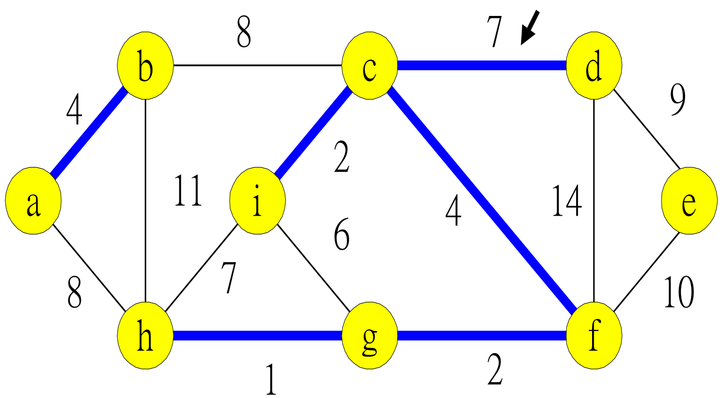

a
b
c
d
e
f
g
h
i
8
7
4
9
11
2
14
6
4
8
7
10
1
2



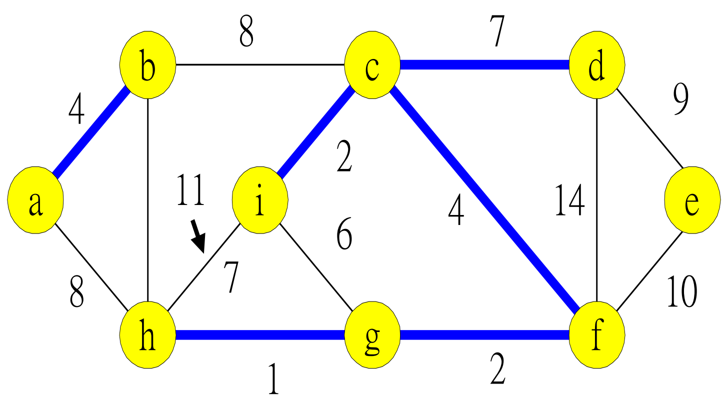
a
b
c
d
e
f
g
h
i
8
7
4
9
11
2
14
4
6
7
8
10
1
2



a
b
c
d
e
f
g
h
i
8
7
4
9
11
2
14
4
6
7
8
10
1
2



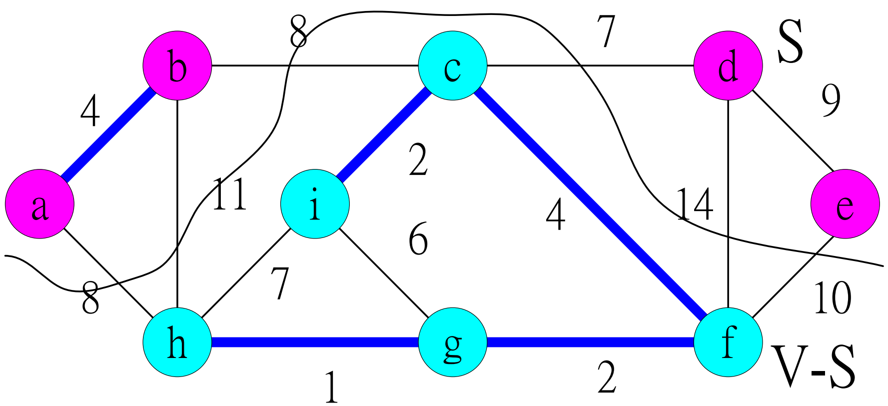
a
b
c
d
e
f
g
h
i
8
7
4
9
11
2
14
4
6
7
8
10
1
2



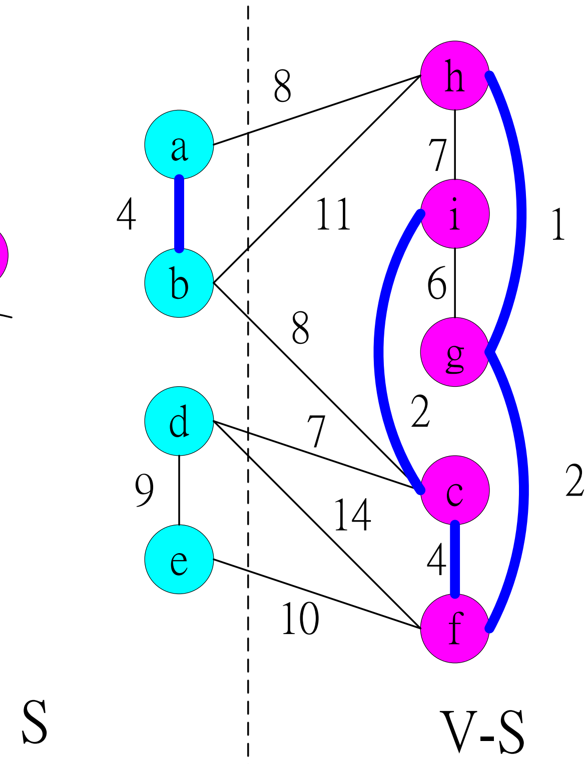
a
b
c
d
e
f
g
h
i
8
7
4
9
11
2
14
4
6
7
8
10
1
2

Continue ......

A ***cut*** (S,V-S) of an undirected graph G = (V,E) is a partition of V. We say that an edge $(u,v) \in E$ ***crosses*** the cut (S,V-S) if one of its endpoints is in S and the other is in V-S. We say a cut ***respects*** the set A of edges if no edge in A crosses the cut. An edge is a ***light edge*** crossing a cut if its weight is the minimum of any edge crossing the cut Note that there can be more than one light edge crossing a cut in case of ties.

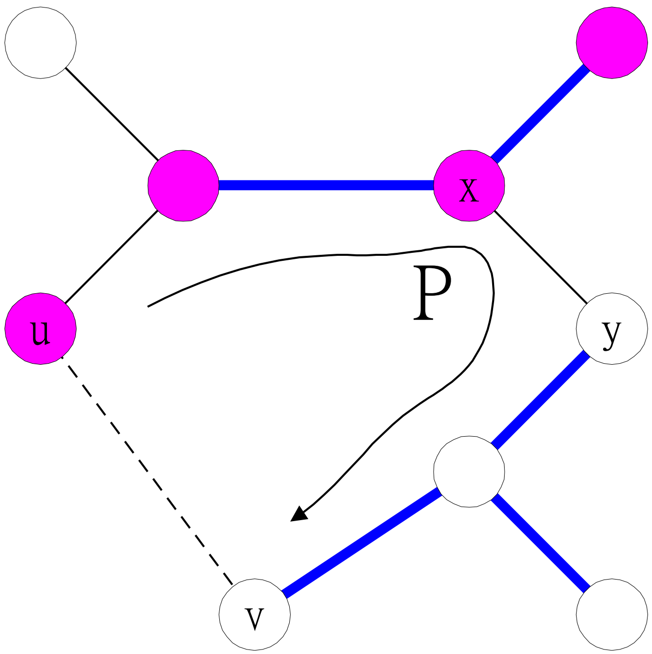


a
b
c
d
e
f
g
h
i
4
8
11
7
2
6
1
2
4
14
7
9
10
8
S
V-S



a
b
d
e
h
i
g
c
f
8
4
11
8
7
9
14
10
7
6
1
2
2
4
S
V-S

Theorem 23.1. Let $G = (V, E)$ be a connected undirected graph with a real-valued weight function w defined on E. Let A be a subset of E that is included in some minimum spanning tree for G, let $(S, V\text{-}S)$ be any cut of G that respects A, and let (u, v) be a light edge crossing $(S, V\text{-}S)$. Then, edge (u, v) is safe for A.



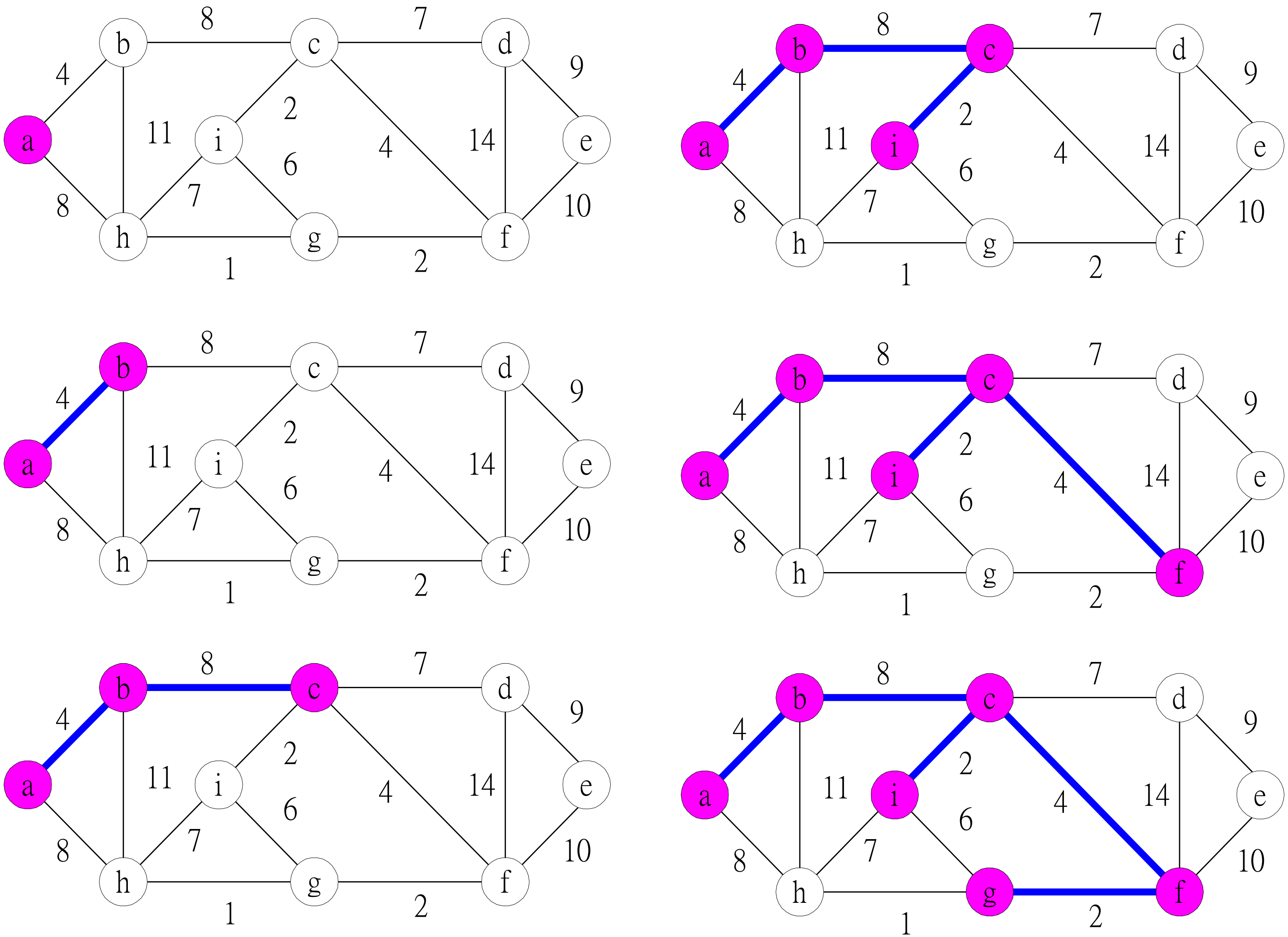
Omit proof of theorem 23.1 and corollary 23.2

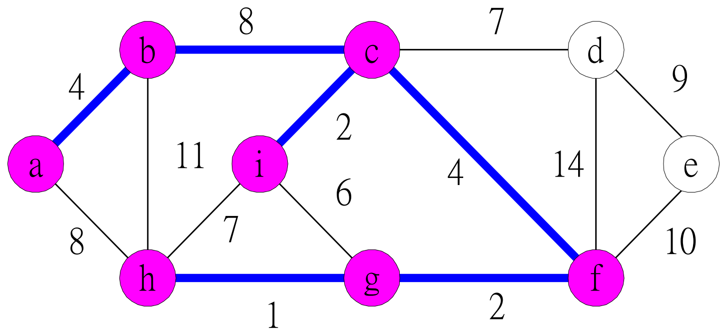
Prim's algorithm

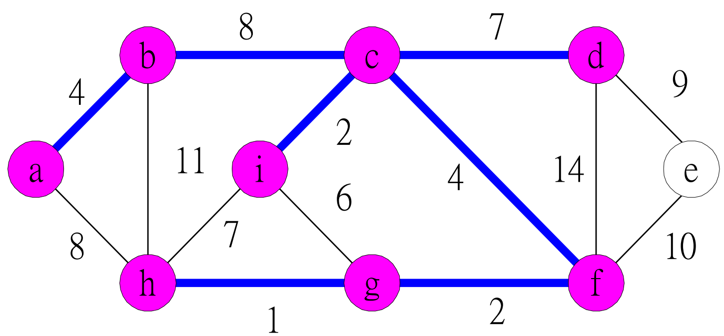
```
MST-PRIM(G,w,r)
For each u € G.V
        u.key =  ∞
        u.π = NIL
r.Key = 0
Q = G.V
While Q ≠ Ǿ
        u = EXTRACT-Min(Q)
        for each v € G.Adj[u]
                if v € Q and w(u,v) < v.key
                        then v.π = u
                        v.key = w(u,v)
```

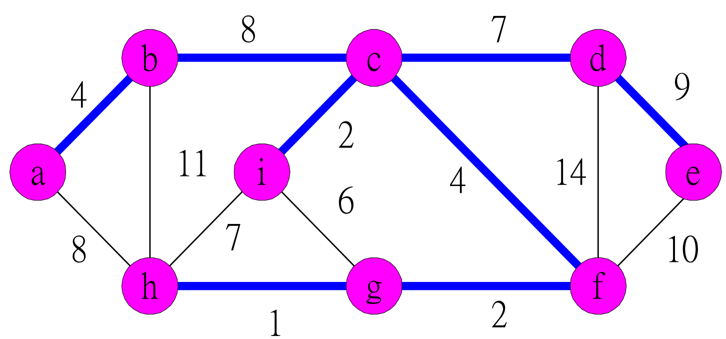
Reading Assignment: Read and understand this complexity calculation from text p. 636

Complexity:
$O(V \log V + E \log V)$, or
$O(E \log V)$


a b c d e f g h i
4 8 7 9 11 2 4 14 6 7 8 10 1 2


a
b
c
d
e
f
g
h
i
4
8
7
9
11
2
6
14
4
8
7
10
1
2


a
b
c
d
e
f
g
h
i
4
8
7
9
11
2
6
14
4
8
7
10
1
2


a
b
c
d
e
f
g
h
i
4
8
7
9
11
2
6
14
4
8
7
10
1
2