CMPN302: Algorithms Design and Analysis

Lecture 08: Minimum Spanning Trees

Ahmed Hamdy

Computer Engineering Department

Cairo University

Fall 2017

Definition

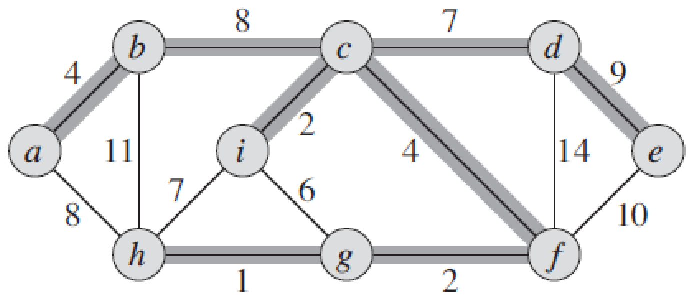


Figure 23.1 A minimum spanning tree for a connected graph. The weights on edges are shown, and the edges in a minimum spanning tree are shaded. The total weight of the tree shown is 37. This minimum spanning tree is not unique: removing the edge (b, c) and replacing it with the edge (a, h) yields another spanning tree with weight 37.

- What is the use of this?!!
  - In electronic circuit design, we need to wire the electric components together

Definition

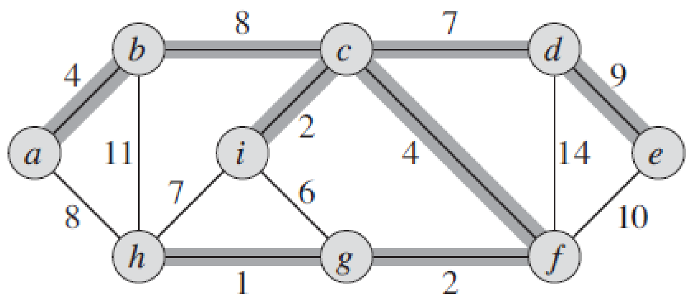


Figure 23.1 A minimum spanning tree for a connected graph. The weights on edges are shown, and the edges in a minimum spanning tree are shaded. The total weight of the tree shown is 37. This minimum spanning tree is not unique: removing the edge (b, c) and replacing it with the edge (a, h) yields another spanning tree with weight 37.

- How to write it as a definition for the problem?
  - Find an acyclic subset $T \subseteq E$ that connects all the vertices with minimum $w(T) = \sum_{(u,v)\in T} w(u,v)$

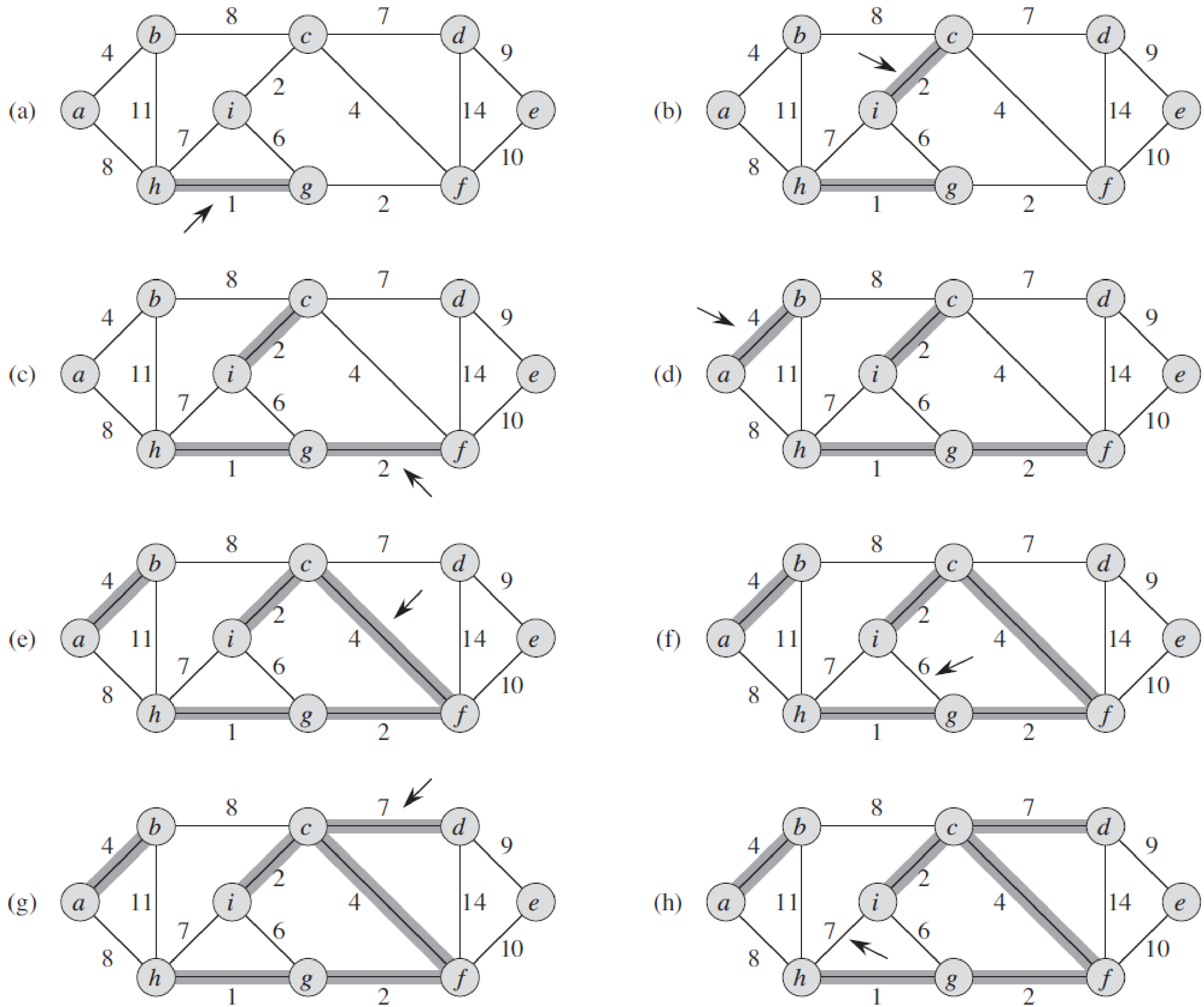
Main concept

GENERIC-MST(G, w)

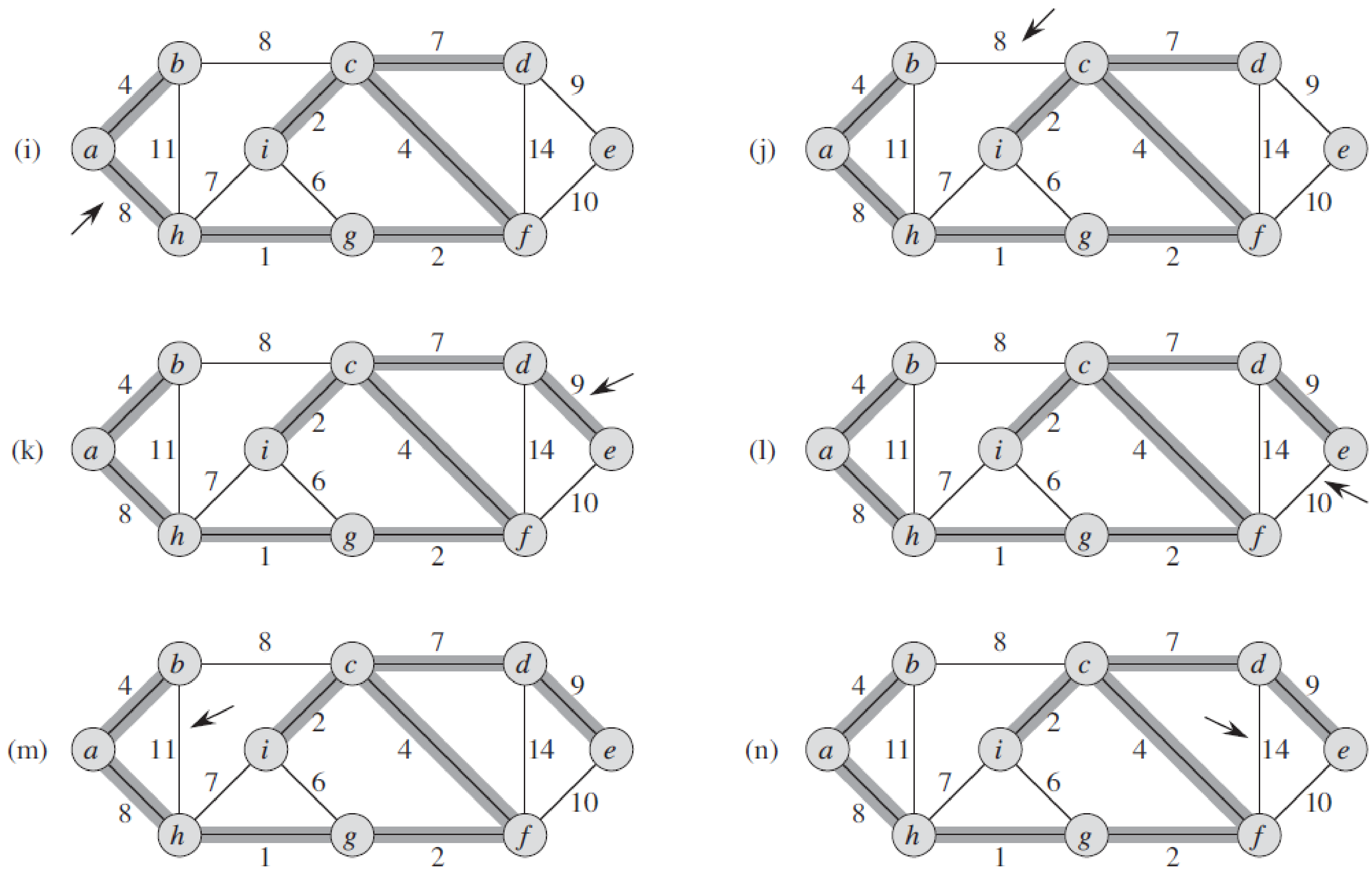
$A = \emptyset$
while A does not form a spanning tree
    find an edge (u, v) that is safe for A
    $A = A \cup \{(u, v)\}$
return A

- Follows which approach??
  - Greedy approach

Kruskal’s algorithm



Kruskal’s algorithm



- Each iteration: a) have a forest and b) add the least-weight safe edge connecting two different components

Kruskal's algorithm

- Algorithm:

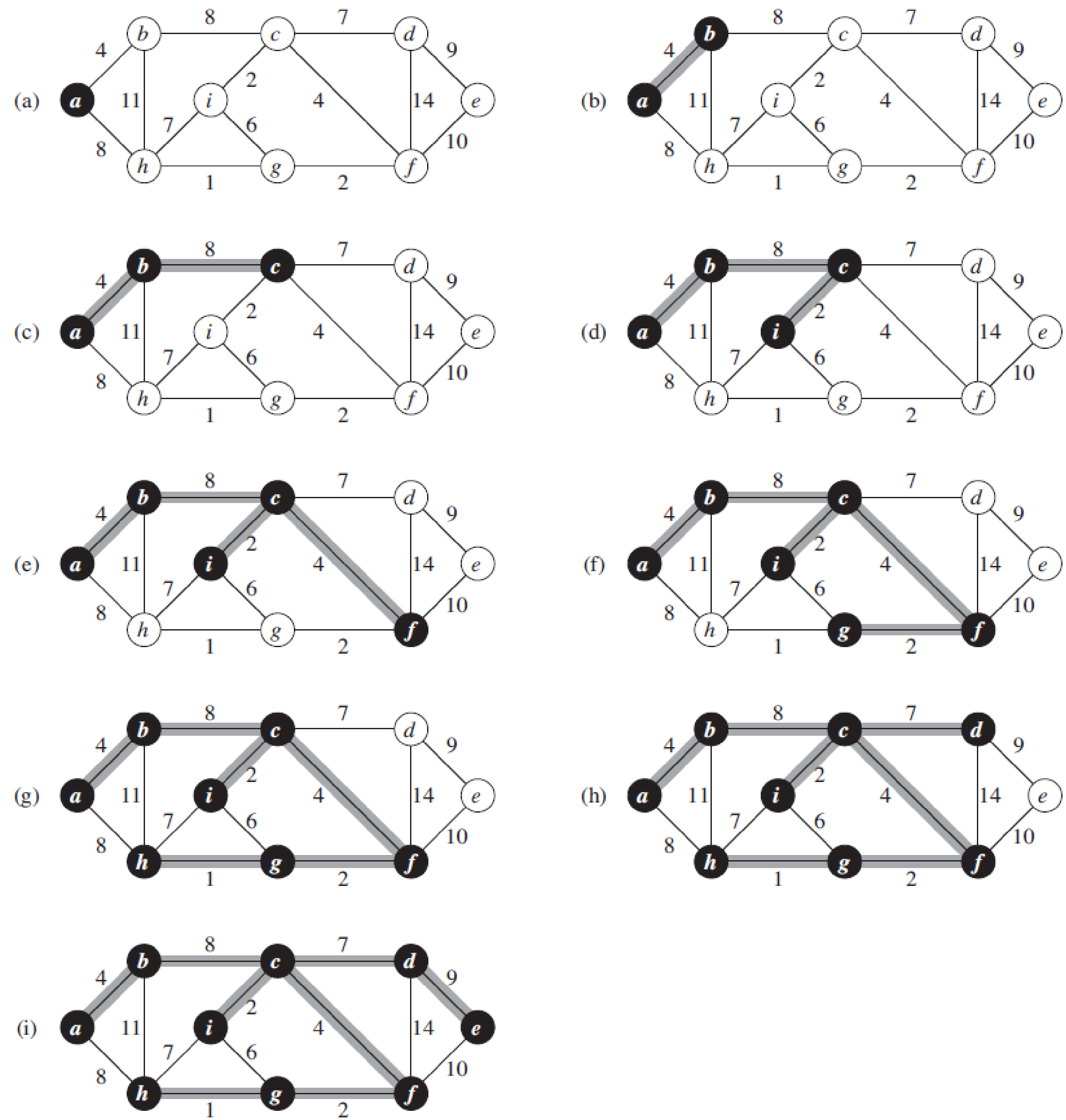
MST-KRUSKAL(G, w)

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

- Complexity: $O(E \log E) = O(E \log V)$

Prim's algorithm

- During each iteration:
  a) have a tree
  b) add the least-weight safe edge connecting the tree to vertex not in tree



Prim's algorithm

- Algorithm:

```
                 MST-PRIM(G, w, r)
O(V)       →   1  for each u ∈ G.V
               2      u.key = ∞
               3      u.π = NIL
               4  r.key = 0
               5  Q = G.V
O(V)       →   6  while Q ≠ ∅
O(log V)   →   7      u = EXTRACT-MIN(Q)
O(E)       →   8      for each v ∈ G.Adj[u]
Lines 6 – 8    9          if v ∈ Q and w(u, v) < v.key
              10              v.π = u
O(log V)   →  11              v.key = w(u, v)
```

- Complexity: $O(V \log V + E \log V) = O(E \log V)$
  - Using Fibonacci heaps: $O(E + V \log V)$