CST 405
Algorithm Analysis & Design

Al Lake
Oregon Institute of Technology

Chapter 23

Minimum Spanning Trees

Minimum Spanning Tree

- **Given a connected undirected graph *G* = (*V*, *e*) and a length function ω such that ω(*e*) is the (positive) length of edge e, find a subset of the edges that connects all of the vertices together and has minimum total length.**
- **The term "length" refers to the original edge weights for the graph, reserving the term "weight" to refer to the weights in the associated matroid.**
- **It is the problem of determining the spanning tree, *T*.**

Minimum Spanning Tree

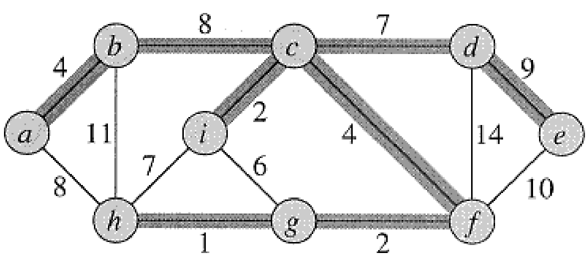

Figure 23.1 A minimum spanning tree for a connected graph. The weights on edges are shown, and the edges in a minimum spanning tree are shaded. The total weight of the tree shown is 37. This minimum spanning tree is not unique: removing the edge (b, c) and replacing it with the edge (a, h) yields another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

Safe Edge

- **A greedy strategy is captured by the following "generic" algorithm, which grows the minimum spanning tree one edge at a time. The algorithm manages a set of edges *A*, maintaining the following loop invariant:**
- **Prior to each iteration, *A* is a subset of some minimum spanning tree.**
- **At each step, an edge (*u*, *v*) is determined which can be added to *A* without violating this invariant, in the sense that $A \cup \{(u, v)\}$ is also a subset of a minimum spanning tree. Such an edge is called a *safe edge* for *A*, since it can be safely added to *A* while maintaining the invariant.**

Graph Cut

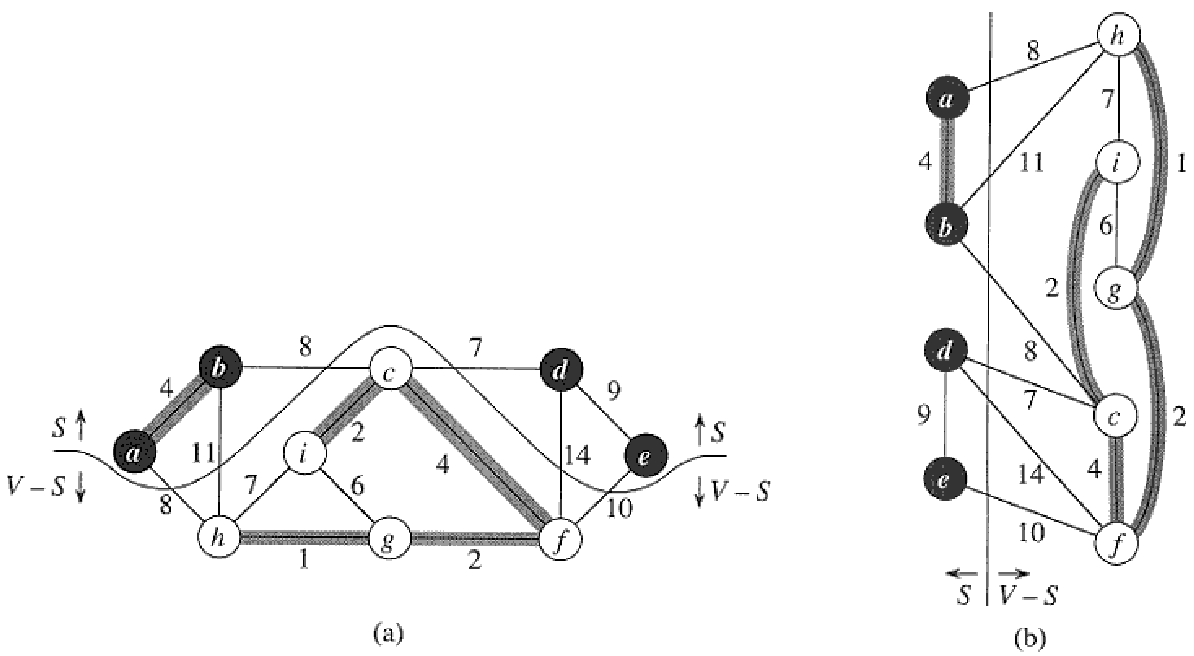

Figure 23.2 Two ways of viewing a cut $(S, V - S)$ of the graph from Figure 23.1. **(a)** The vertices in the set S are shown in black, and those in $V - S$ are shown in white. The edges crossing the cut are those connecting white vertices with black vertices. The edge (d, c) is the unique light edge crossing the cut. A subset A of the edges is shaded; note that the cut $(S, V - S)$ respects A, since no edge of A crosses the cut. **(b)** The same graph with the vertices in the set S on the left and the vertices in the set $V - S$ on the right. An edge crosses the cut if it connects a vertex on the left with a vertex on the right.

Kruskal

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

Prim

```
MST-PRIM(G, w, r)
for each u ∈ V[G]
      do key[u] ← ∞
         π[u] ← NIL
key[r] ← 0
Q ← V[G]
while Q ≠ Ø
      do u ← EXTRACT-MIN(Q)
         for each v ∈ Adj[u]
             do if v ∈ Q and w(u, v) < key[v]
                   then π[v] ← u
                        key[v] ← w(u, v)
```

Prim

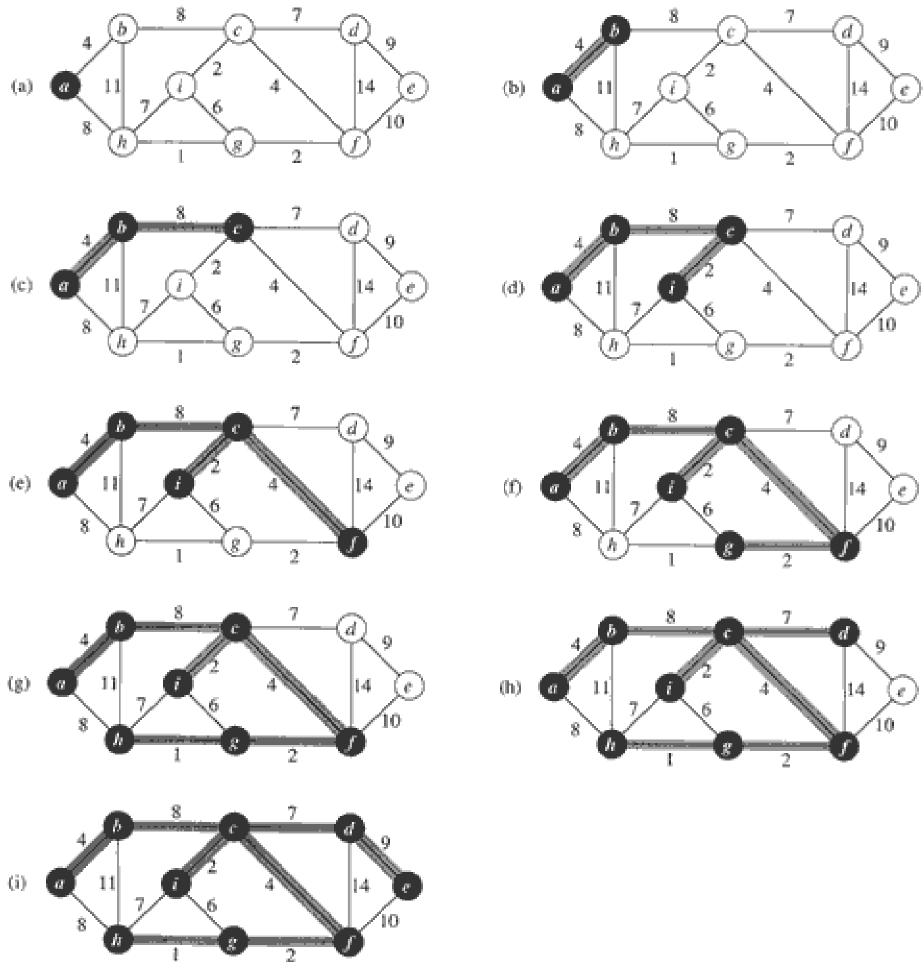

Figure 23.5 The execution of Prim's algorithm on the graph from Figure 23.1. The root vertex is a. Shaded edges are in the tree being grown, and the vertices in the tree are shown in black. At each step of the algorithm, the vertices in the tree determine a cut of the graph, and a light edge crossing the cut is added to the tree. In the second step, for example, the algorithm has a choice of adding either edge (b, c) or edge (a, h) to the tree since both are light edges crossing the cut.