PRIM’SALGORITHM

A ***spanning tree*** of an undirected connected graph is its connected acyclic subgraph (i.e., a tree) that contains all the vertices of the graph. If such a graph has weights assigned to its edges, a ***minimum spanning tree*** is its spanning tree of the smallest weight, where the ***weight*** of a tree is defined as the sum of the weights on all its edges. The ***minimum spanning tree problem*** is the problem of finding a minimum spanning tree for a given weighted connected graph.

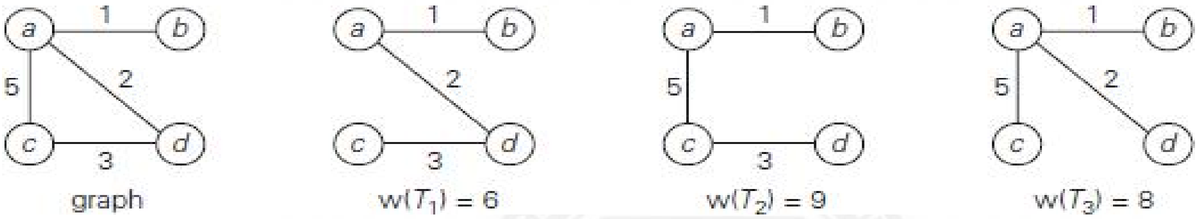


FIGURE 3.13 Graph and its spanning trees, with *T*1 being the minimum spanning tree.

The minimum spanning tree is illustrated in Figure 3. If we were to try constructing a minimum spanning tree by exhaustive search, we would face two serious obstacles. First, the number of spanning trees grows exponentially with the graph size (at least for dense graphs). Second, generating all spanning trees for a given graph is not easy; in fact, it is more difficult than finding a minimum spanning tree for a weighted graph.

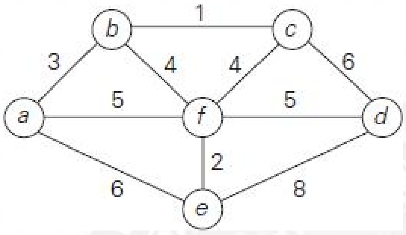
Prim’s algorithm constructs a minimum spanning tree through a sequence of expanding subtrees. The initial subtree in such a sequence consists of a single vertex selected arbitrarily from the set V of the graph’s vertices. On each iteration, the algorithm expands the current tree in the greedy manner by simply attaching to it the nearest vertex not in that tree. The algorithm stops after all the graph’s vertices have been included in the tree being constructed.

ALGORITHM *Prim(G)*

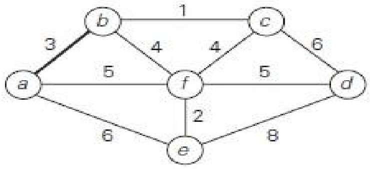
```
    //Prim’s algorithm for constructing a minimum spanning tree
    //Input: A weighted connected graph $G = \{V, E\}$
    //Output: $E_T$, the set of edges composing a minimum
    spanning tree of $G$ $V_T \leftarrow \{v_0\}$ //the set of tree vertices can be
    initialized with any vertex $E_T \leftarrow \Phi$
    for $i \leftarrow 1$ to $|V| - 1$ do
        findaminimum-
        weightedgee□=(v□,u□)amongalltheedges(v,u) such that $v$
```

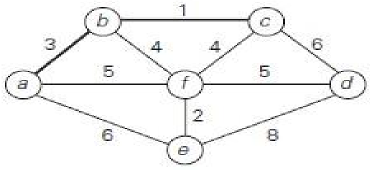
```
            is in $V_T$ and u is in V −$V_T$
            $V_T \leftarrow V_T \cup \{u^*\}$
            $E_T \leftarrow E_T \cup \{e^*\}$
    return $E_T$
```

If a graph is represented by its adjacency lists and the priority queue is implemented as a min-heap, the running time of the algorithm is O(|E| log |V |) in a connected graph, where |V| − 1≤
|E|.



Tree vertices	Remaining vertices	Illustration

a(−, −)	**b(a, 3)** c(−, ∞) d(−, ∞)
	e(a, 6) f(a, 5)
		

b(a, 3)	**c(b, 1)** d(−, ∞) e(a, 6)
	f(b, 4)
		

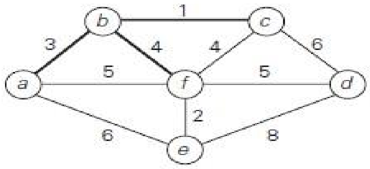
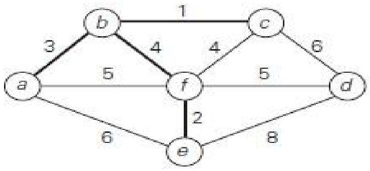
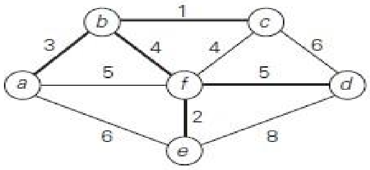
c(b, 1)	d(c, 6) e(a, 6) **f(b, 4)**	
f(b, 4)	d(f, 5) **e(f, 2)**	
e(f, 2)	**d(f, 5)**	
d(f, 5)		

FIGURE Application of Prim's algorithm. The parenthesized labels of a vertex in the middle
column indicate the nearest tree vertex and edge weight; selected vertices and edges are in bold.