Network definitions. A network consists of a set of **nodes** linked by **arcs** (or **branches**). The notation for describing a network is (N, A), where N is the set of nodes, and A is the set of arcs. As an illustration, the network in Figure 6.1 is described as

$$N = \{1, 2, 3, 4, 5\}$$
$$A = \{(1, 2), (1, 3), (2, 3), (2, 5), (3, 4), (3, 5), (4, 2), (4, 5)\}$$

Associated with each network is a **flow** (e.g., oil products flow in a pipeline and automobile traffic flow in highways). The maximum flow in a network can be finite or infinite, depending on the capacity of its arcs.

An arc is said to be **directed** or **oriented** if it allows positive flow in one direction only. A **directed network** has all directed arcs.

A **path** is a set of arcs joining two distinct nodes, passing through other nodes in the network. For example, in Figure 6.1, arcs $(1, 2)$, $(2, 3)$, $(3, 4)$, and $(4, 5)$ form a path between nodes 1 and 5. A path forms a **cycle** or a **loop** if it connects a node back to itself through other nodes. In Figure 6.1, arcs $(2, 3)$, $(3, 4)$, and $(4, 2)$ form a cycle.

A network is said to be **connected** if every two distinct nodes are linked by at least one path. The network in Figure 6.1 demonstrates this type of network. A **tree** is a *cycle-free* connected network comprised of a *subset* of all the nodes, and a **spanning tree** links *all* the nodes of the network. Figure 6.2 provides examples of a tree and a spanning tree from the network in Figure 6.1.

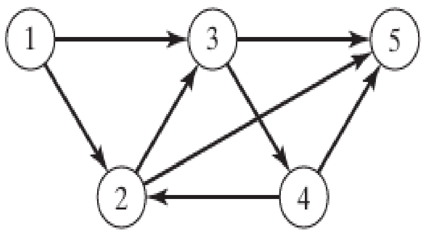
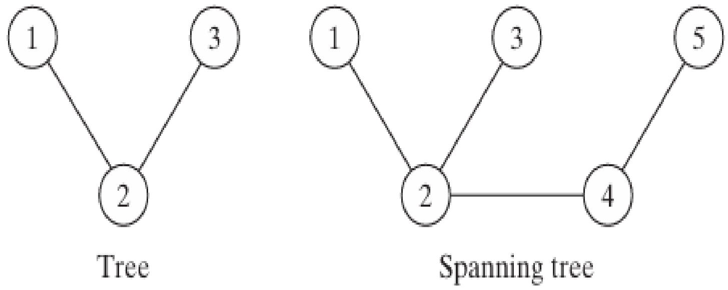


FIGURE 6.1

Example of (N, A) Network

FIGURE 6.2

Examples of a tree and a spanning tree



MINIMAL SPANNING TREE ALGORITHM

The minimal spanning tree links the nodes of a network using the smallest total length of connecting branches. A typical application occurs in the pavement of roads linking towns, either directly or passing through other towns. The minimal spanning tree solution provides the most economical design of the road system.

Let $N = \{1, 2, \ldots, n\}$ be the set of nodes of the network and define

C_k = Set of nodes that have been permanently connected at iteration k

$\overline{C}_k$ = Set of nodes as yet to be connected permanently after iteration k

The following steps describe the minimal spanning tree algorithm:

Step 0. Set $C_0 = \varnothing$ and $\overline{C}_0 = N$.

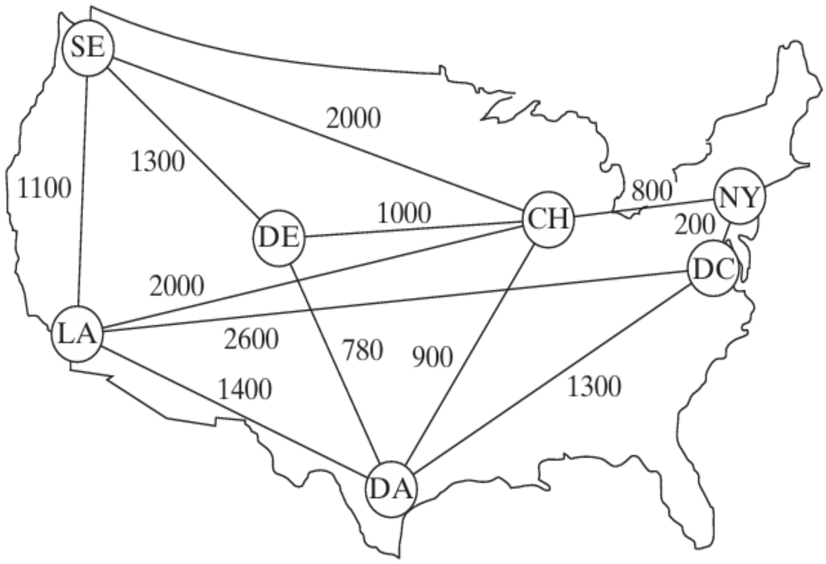
Step 1. Start with *any* node i in the unconnected set $\overline{C}_0$ and set $C_1 = \{i\}$, rendering $\overline{C}_1 = N - \{i\}$. Set $k = 2$.

General step *k*. Select a node, j^*, in the unconnected set $\overline{C}_{k-1}$ that yields the shortest arc to a node in the connected set C_{k-1}. Link j^* permanently to C_{k-1} and remove it from $\overline{C}_{k-1}$ to obtain C_k and $\overline{C}_k$, respectively. Stop if $\overline{C}_k$ is empty; else, set $k = k + 1$ and repeat the step.

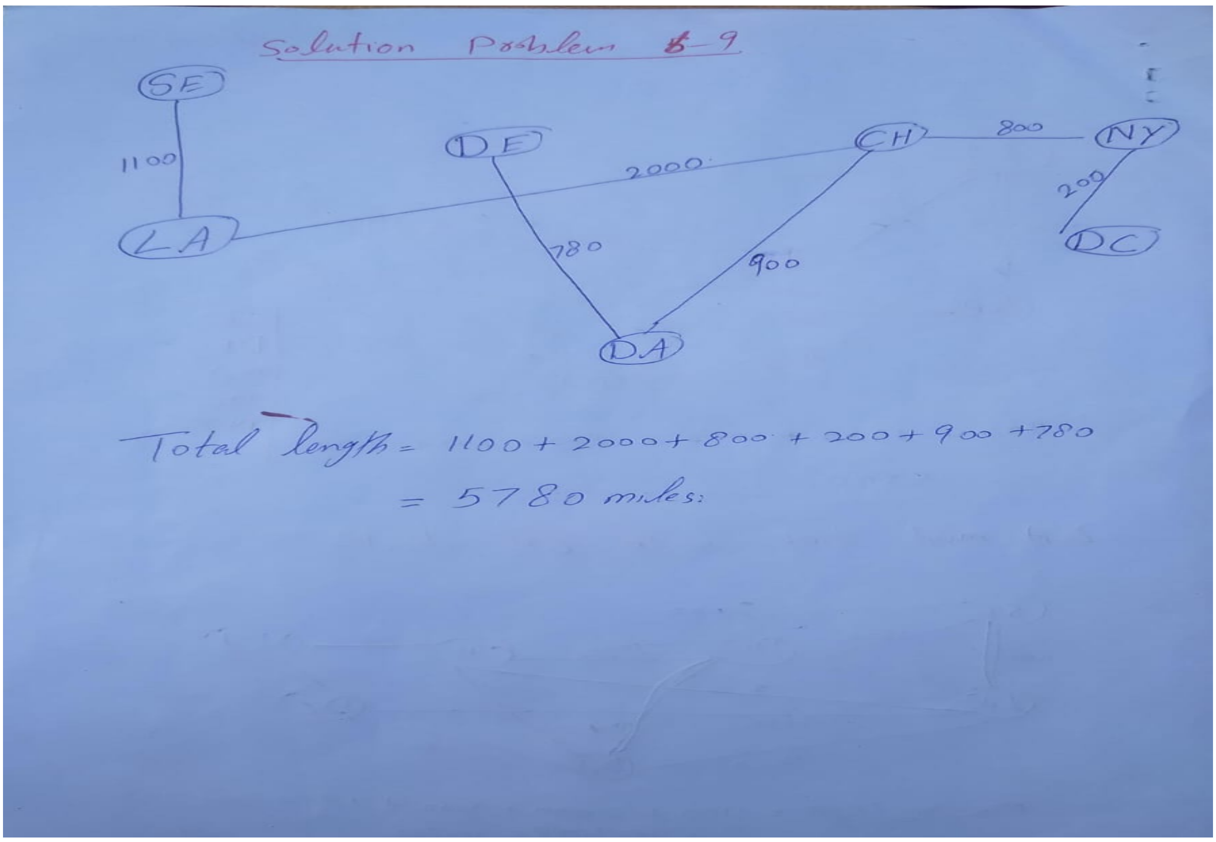
6-9. In intermodal transportation, loaded truck trailers are shipped between railroad terminals on special flatbed carts. Figure 6.33 shows the location of the main railroad terminals in the United States and the existing railroad tracks. The objective is to decide which tracks should be "revitalized" to handle the intermodal traffic. In particular, the Los Angeles (LA) terminal must be linked directly to Chicago (CH) to accommodate expected heavy traffic. Other than that, all the remaining terminals can be linked, directly or indirectly, such that the total length (in miles) of the selected tracks is minimized. Determine the segments of the railroad tracks that must be included in the revitalization program.

FIGURE 6.33

Network for Problem 6-9



Solution:



6-10. Figure 6.34 gives the mileage of the feasible links connecting nine offshore natural gas wellheads with an inshore delivery point. Because wellhead 1 is the closest to shore, it is equipped with sufficient pumping and storage capacity to pump the output of the remaining eight wells to the delivery point. Determine the minimum pipeline network that links the wellheads to the delivery point.

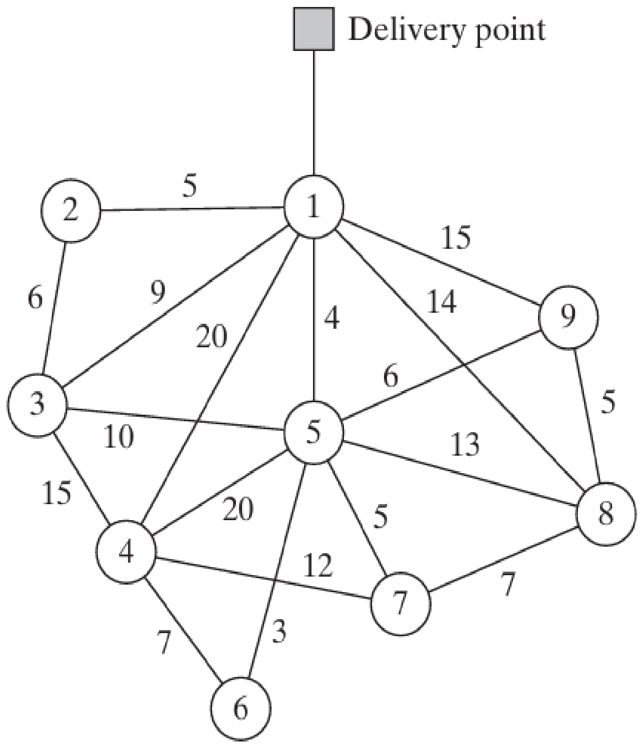
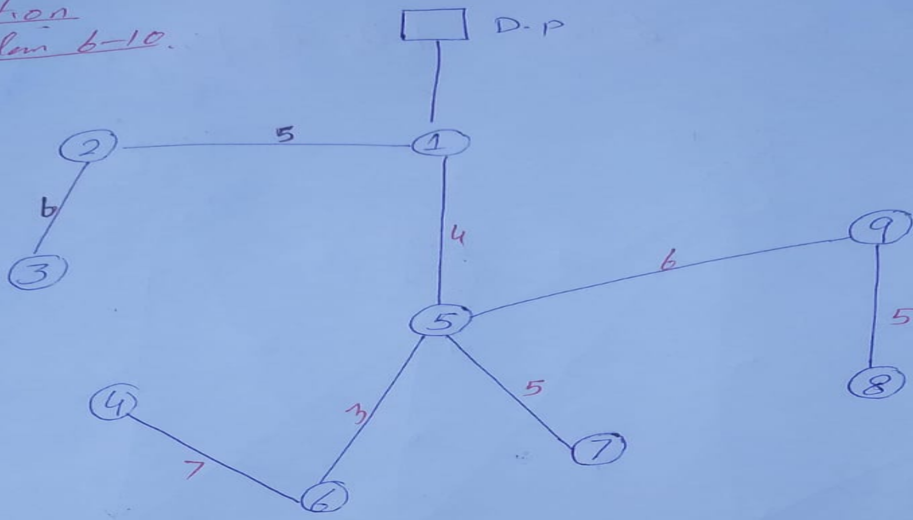


FIGURE 6.34
Network for Problem 6-10 and 6-11

Solution:

Solution

Problem 6-10.



Minimum Pipeline network length = $5+6+4+6+5+5+3+7$

$= 41$ mile

Shortest-Route Algorithms

This section presents two algorithms for solving both cyclic (i.e., containing loops) and acyclic networks:

1. Dijkstra's algorithm for determining the shortest routes between the source node and every other node in the network.
2. Floyd's algorithm for determining the shortest route between *any* two nodes in the network.

Essentially, Floyd's algorithm subsumes Dijkstra's.

Dijkstra's algorithm. Let u_i be the shortest distance from source node 1 to node i, and define d_{ij} (≥ 0) as the length of arc (i, j). The algorithm defines the label for an immediately succeeding node j as

$$[u_j, i] = [u_i + d_{ij}, i], d_{ij} \geq 0$$

The label for the starting node is $[0, —]$, indicating that the node has no predecessor.

Node labels in Dijkstra's algorithm are of two types: *temporary* and *permanent*. A temporary label at a node is modified if a shorter route to the node can be found. Otherwise, the temporary status is changed to permanent.

Step 0. Label the source node (node 1) with the *permanent* label $[0, —]$. Set $i = 1$.

General step *i*.

(a) Compute the *temporary* labels $[u_i + d_{ij}, i]$ for each node j with $d_{ij} > 0$, provided j is not permanently labeled. If node j already has an existing temporary label $[u_j, k]$ via another node k and if $u_i + d_{ij} < u_j$, replace $[u_j, k]$ with $[u_i + d_{ij}, i]$.

(b) If *all* the nodes have *permanent* labels, stop. Otherwise, select the label $[u_r, s]$ having the shortest distance $(= u_r)$ among all the *temporary* labels (break ties arbitrarily). Set $i = r$ and repeat step i.

6-18. The network in Figure 6.36 gives the distances in miles between pairs of cities 1, 2, …, and 8. Use Dijkstra's algorithm to find the shortest route between the following cities:

(a) Cities 1 and 7.

(b) Cities 1 and 6.

*__(c)__ Cities 4 and 8.

(d) Cities 2 and 7.

6-19. Use Dijkstra's algorithm to find the shortest route between node 1 and every other node in the network of Figure 6.37.

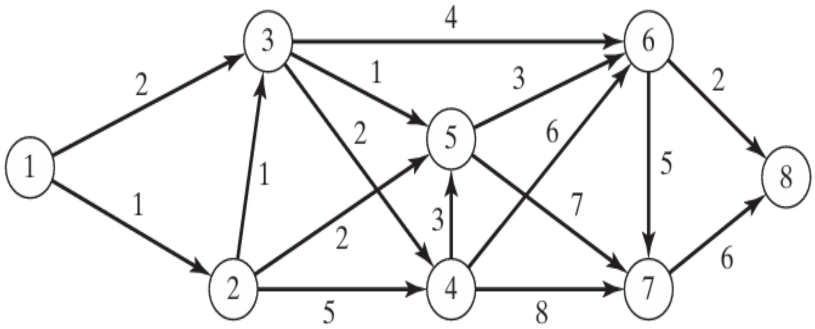


FIGURE 6.36

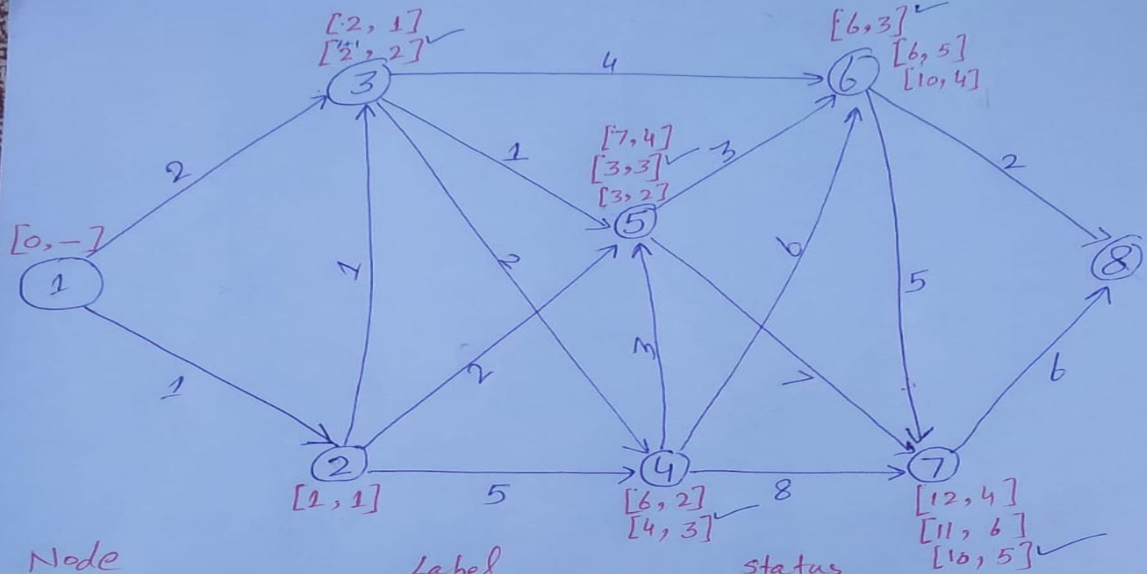
Network for Problem 6-18

Solution:

Solution Problem 6-18

$[U_j, i] = [U_i + d_{ij}, i]$

Distance   Node



Node	Label	Status
1	[0, –]	Permanent
2	[1, 1]	〃
3	[1+1, 2], [2, 1]	[2, 2]
4	[6, 2], [4, 3]	[4, 3]
5	[3, 2], [3, 3], [7, 4]	[3, 3]
6	[6, 3], [6, 5], [10, 4]	[6, 3]
7	[12, 4], [11, 6], [10, 5]	[10, 5]

Total Distance = 10

Alternative routes

i) 1 → 3 → 5 → 7

ii) 1 → 2 → 3 → 5 → 7

iii) 1 → 2 → 5 → 7