The Design and Analysis of Algorithm

Fundamental Data Structures

Instructor: Sadullah Karimi

M.Sc. in CSE

Linear Data Structures

- The two most important elementary data structures are the array and the linked list.
- A (one-dimensional) array is a sequence of n items of the same data type that are stored contiguously in computer memory and made accessible by specifying a value of the array's index

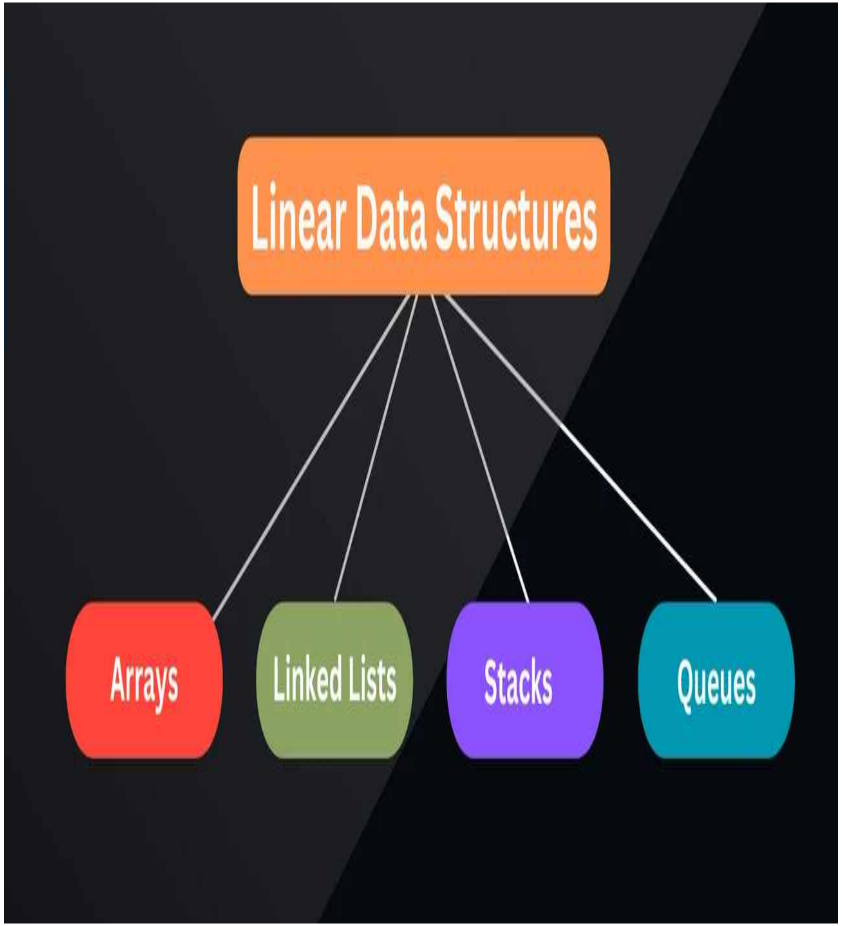


FIGURE 1.3 Array of n elements.

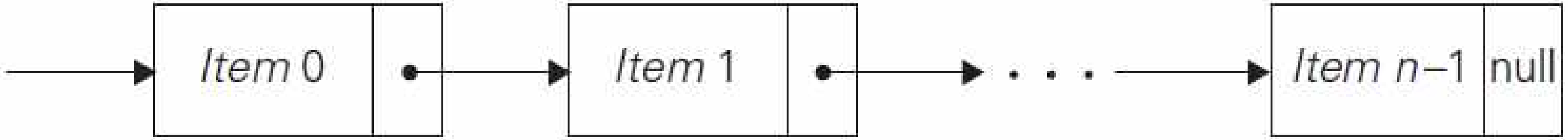
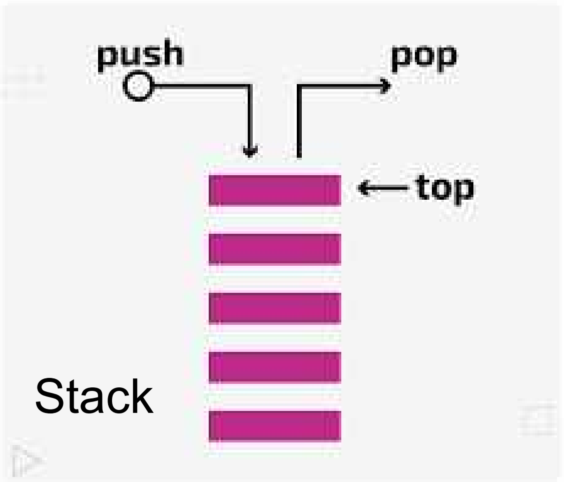


FIGURE 1.4 Singly linked list of n elements.

Stack, Queue

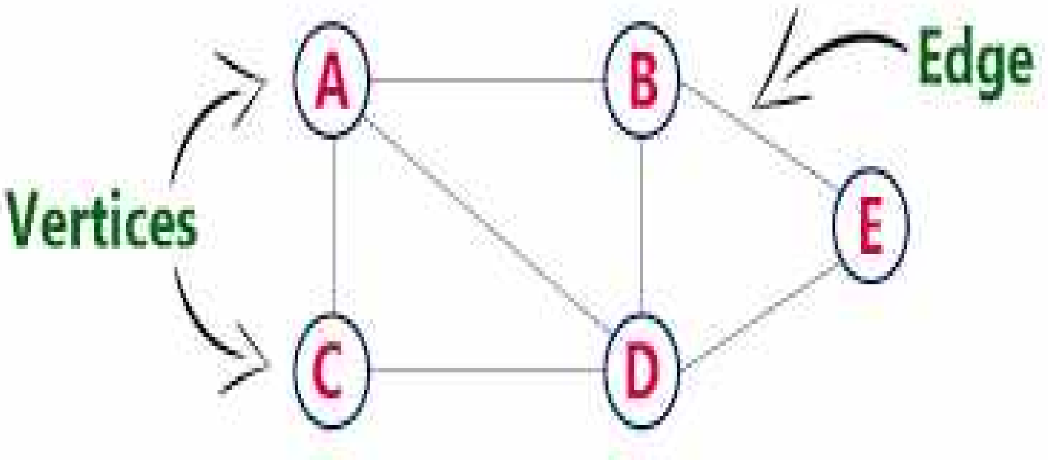
- A stack is a list in which insertions and deletions can be done only at the end.
- A queue, on the other hand, is a list from which elements are deleted from one end of the structure, called the front (this operation is called dequeue), and new elements are added to the other end, called the rear (this operation is called enqueue).



Graph

- As we mentioned in the previous section, a graph is informally thought of as a collection of points in the plane called “vertices” or “nodes,” some of them connected by line segments called “edges” or “arcs.”

- Formally, a graph G = V,E is defined by a pair of two sets: a finite nonempty set V of items called vertices and a set E of pairs of these items called edges



$$V = \{a, b, c, d, e, f\},\ E = \{(a, c), (a, d), (b, c), (b, f), (c, e), (d, e), (e, f)\}.$$

The digraph depicted in Figure 1.6b has six vertices and eight directed edges:

$$V = \{a, b, c, d, e, f\},\ E = \{(a, c), (b, c), (b, f), (c, e), (d, a), (d, e), (e, c), (e, f)\}.$$

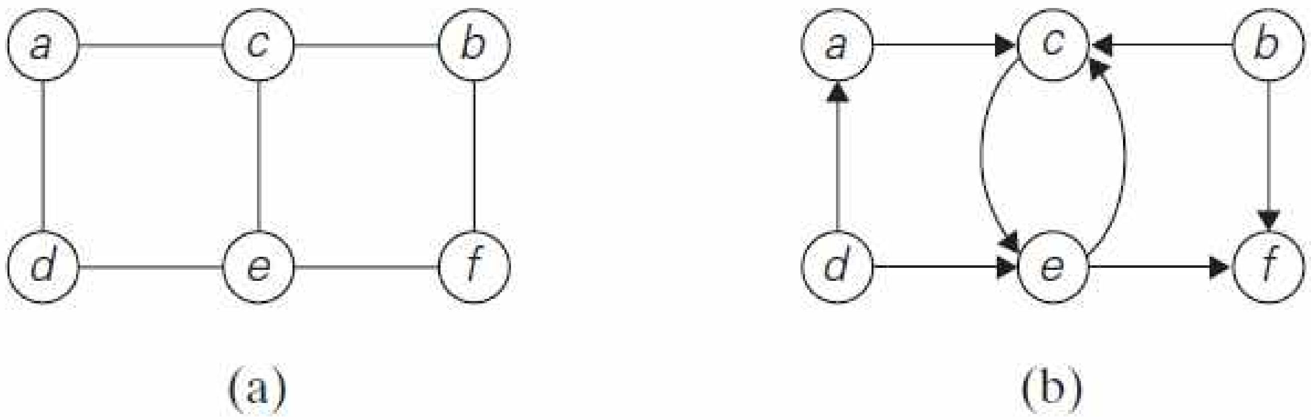


FIGURE 1.6 (a) Undirected graph. (b) Digraph.

Graph Representations

- Graphs for computer algorithms are usually represented in one of two ways:
- The adjacency matrix and adjacency lists.
- The adjacency matrix of a graph with n vertices is an n × n Boolean matrix with one row and one column for each of the graph's vertices, in which the element in the ith row and the jth column is equal to 1 if there is an edge from the ith vertex to the jth vertex, and equal to 0 if there is no such edge.
- The adjacency lists of a graph or a digraph is a collection of linked lists, one for each vertex, that contain all the vertices adjacent to the list's vertex (i.e., all the vertices connected to it by an edge). Usually, such lists start with a header identifying a vertex for which the list is compiled.

$$
\begin{array}{cc}
 & \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \end{array} &
\begin{bmatrix}
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0
\end{bmatrix}
\end{array}
$$

(a)

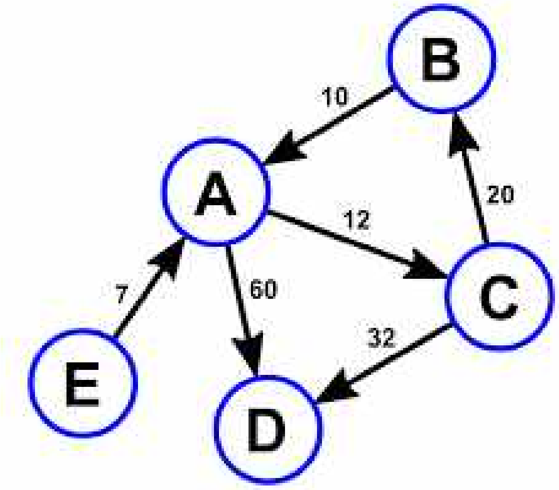
$$
\begin{array}{|c|l}
\hline
a & \rightarrow c \rightarrow d \\
\hline
b & \rightarrow c \rightarrow f \\
\hline
c & \rightarrow a \rightarrow b \rightarrow e \\
\hline
d & \rightarrow a \rightarrow e \\
\hline
e & \rightarrow c \rightarrow d \rightarrow f \\
\hline
f & \rightarrow b \rightarrow e \\
\hline
\end{array}
$$

(b)

FIGURE 1.7 (a) Adjacency matrix and (b) adjacency lists of the graph in Figure 1.6a.

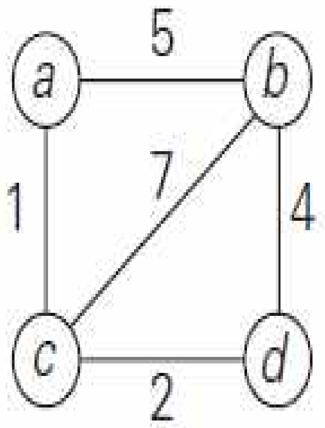
Weighted Graphs

- A weighted graph (or weighted digraph) is a graph (or digraph) with numbers assigned to its edges. These numbers are called weights or costs.
- An interest in such graphs is motivated by numerous real-world applications, such as finding the shortest path between two points in a transportation or communication network or the traveling salesman problem mentioned earlier.



Paths and Cycles

- Among the many properties of graphs, two are important for a great number of applications: connectivity and acyclicity. Both are based on the notion of a path.Apath from vertex u to vertex v of a graph G can be defined as a sequence of adjacent (connected by an edge) vertices that starts with u and ends with v.
- If all vertices of a path are distinct, the path is said to be simple.
- The length of a path is the total number of vertices in the vertex sequence defining the path minus 1, which is the same as the number of edges in the path. For example, a, c, b, f is a simple path of length 3 from a to f in the graph in Figure 1.6a, whereas a, c, e, c, b, f is a path (not simple) of length 5 from a to f.



(a)

$$\begin{array}{c c} & \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} \infty & 5 & 1 & \infty \\ 5 & \infty & 7 & 4 \\ 1 & 7 & \infty & 2 \\ \infty & 4 & 2 & \infty \end{bmatrix} \end{array}$$

(b)

a	$\rightarrow b, 5 \rightarrow c, 1$
b	$\rightarrow a, 5 \rightarrow c, 7 \rightarrow d, 4$
c	$\rightarrow a, 1 \rightarrow b, 7 \rightarrow d, 2$
d	$\rightarrow b, 4 \rightarrow c, 2$

(c)

FIGURE 1.8 (a) Weighted graph. (b) Its weight matrix. (c) Its adjacency lists.

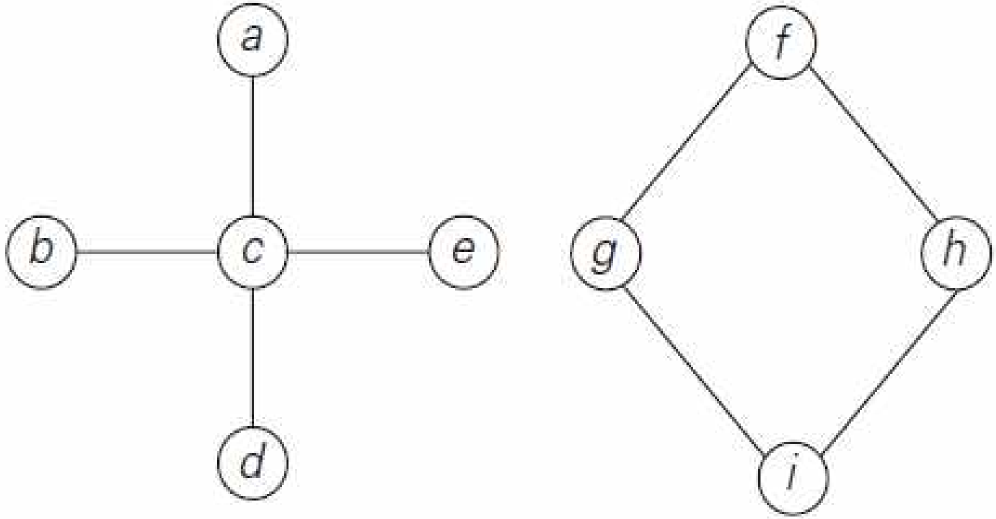


FIGURE 1.9 Graph that is not connected.

Trees

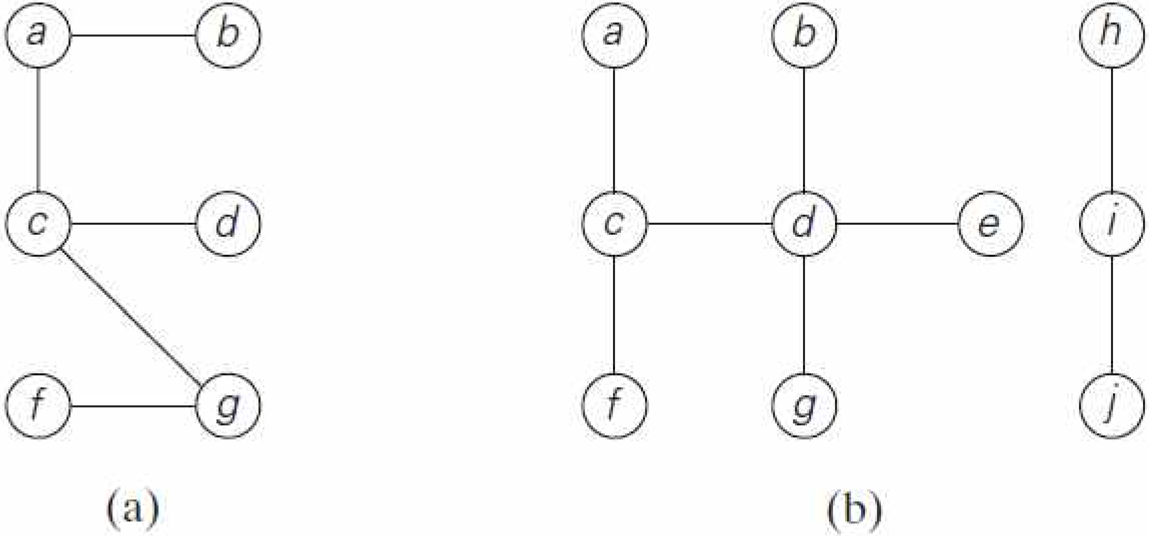


FIGURE 1.10 (a) Tree. (b) Forest.

Binary Tree

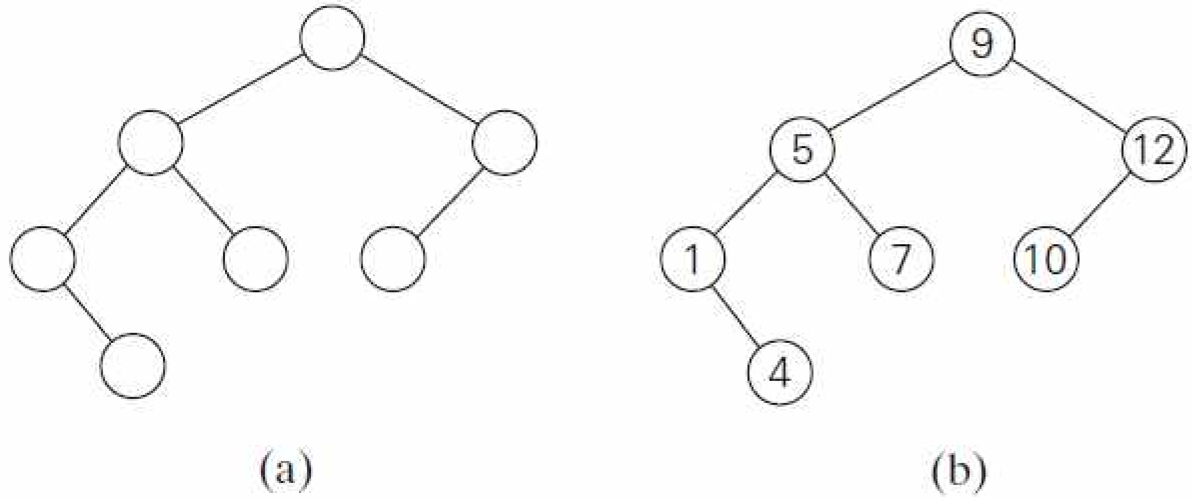


FIGURE 1.12 (a) Binary tree. (b) Binary search tree.

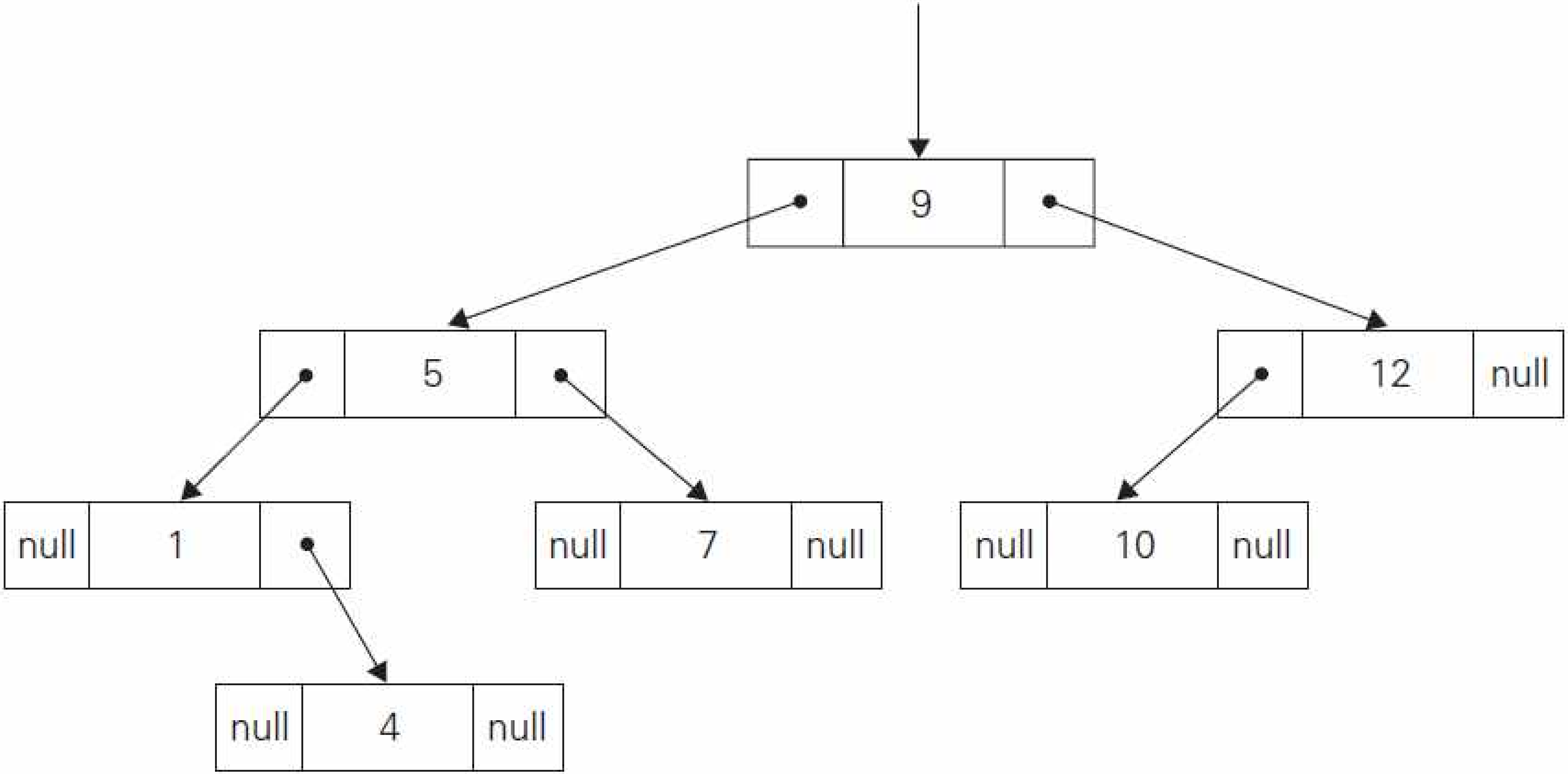


FIGURE 1.13 Standard implementation of the binary search tree in Figure 1.12b.

Sets and Dictionaries

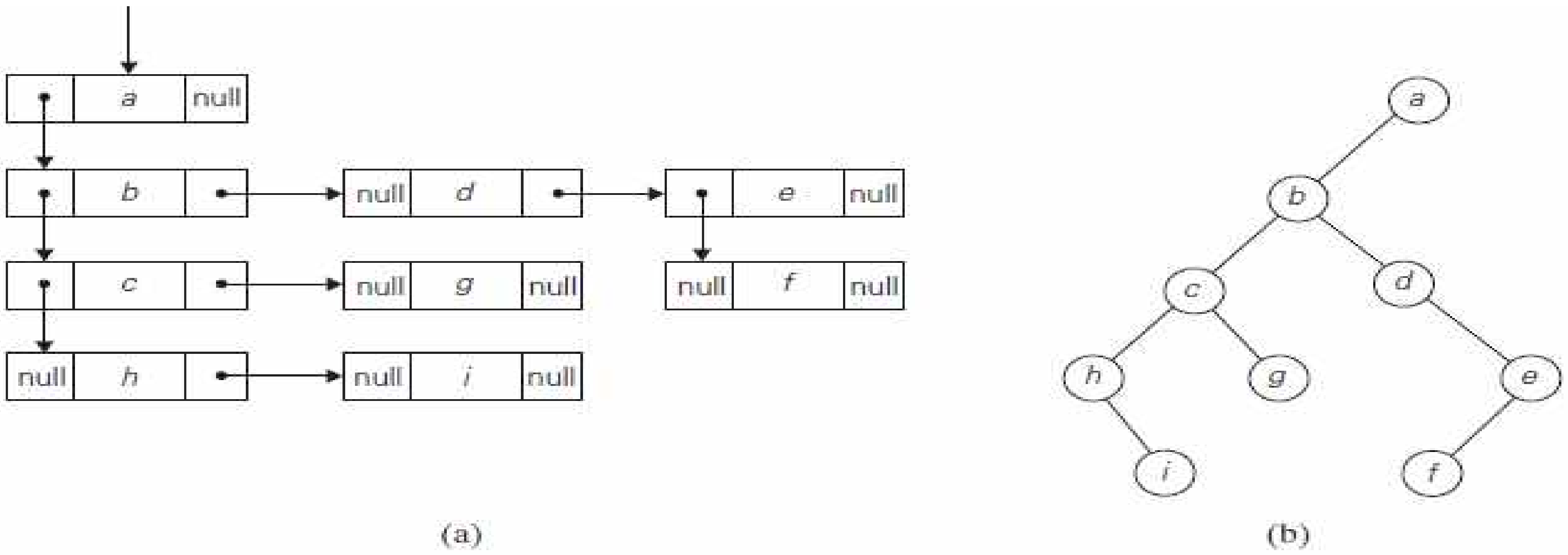


FIGURE 1.14 (a) First child–next sibling representation of the tree in Figure 1.11b. (b) Its binary tree representation.

Thanks for your attention

Any questions are appreciated