SNS COLLEGE OF TECHNOLOGY

Coimbatore-35.

An Autonomous Institution

COURSE NAME : 19ITB201   DESIGN AND ANALYSIS OF ALGORITHMS

II YEAR/ IV SEMESTER

UNIT – I INTRODUCTION

1.4 IMPORTANT PROBLEM TYPES

Important Problem types

- Sorting
- Searching
- String processing
- Graph problems
- Combinatorial problems
- Geometric problems
- Numerical problems

Fundamental data structures

A **data structure** can be defined as a particular scheme of organizing related data items.

a. Linear data structures

b. Graphs

c. Trees

d. Sets and dictionaries

Trees

- A ***tree*** (more accurately, a ***free tree***) is a connected acyclic graph.
- A graph with no cycles is said to be ***acyclic***.

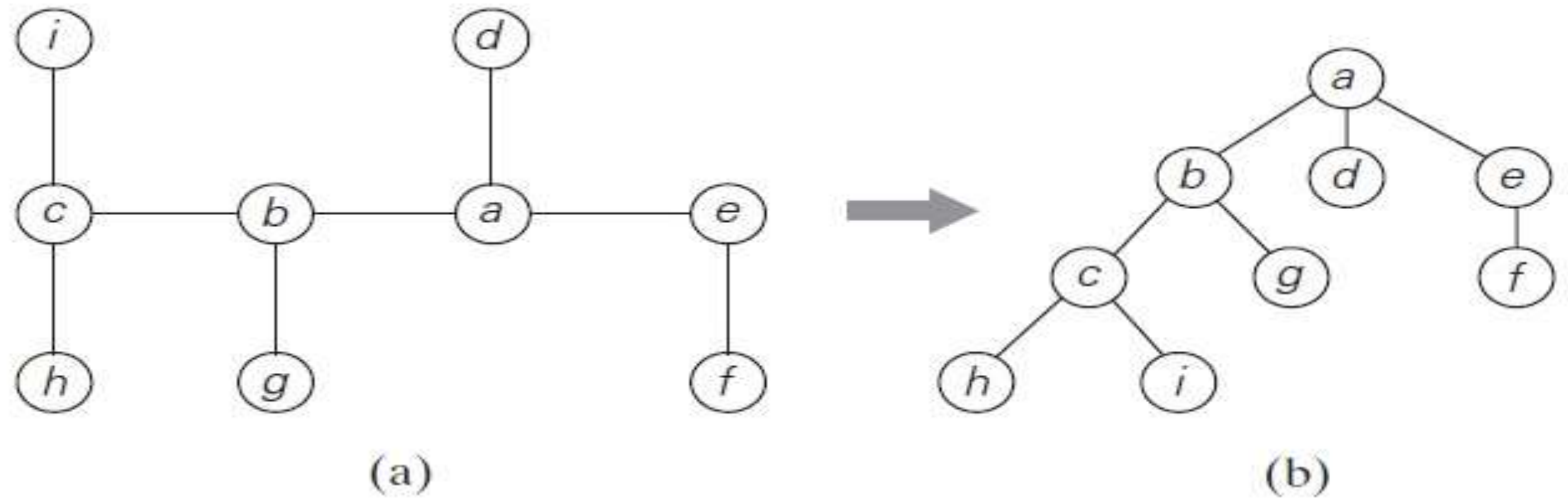


FIGURE 1.11 (a) Free tree. (b) Its transformation into a rooted tree.

Trees

Rooted Trees :

- Another very important property of trees is the fact that for every two vertices in a tree, there always exists exactly one simple path from one of these vertices to the other. This property makes it possible to select an arbitrary vertex in a free tree and consider it as the ***root*** of the so-called ***rooted tree***.
- A rooted tree is usually depicted by placing its root on the top (level 0 of the tree), the vertices adjacent to the root below it (level 1), the vertices two edges apart from the root still below (level 2), and so on. Figure 1.11 presents such a transformation from a free tree to a rooted tree.

Trees

Ordered Trees :

- An **ordered tree** is a rooted tree in which all the children of each vertex are ordered. It is convenient to assume that in a tree's diagram, all the children are ordered left to right.
- A **binary tree** can be defined as an ordered tree in which every vertex has no more than two children and each child is designated as either a **left child** or a **right child** of its parent; a binary tree may also be empty.
- The binary tree with its root at the left (right) child of a vertex in a binary tree is called the **left** (**right**) **subtree** of that vertex.

Example of Binary Search Trees

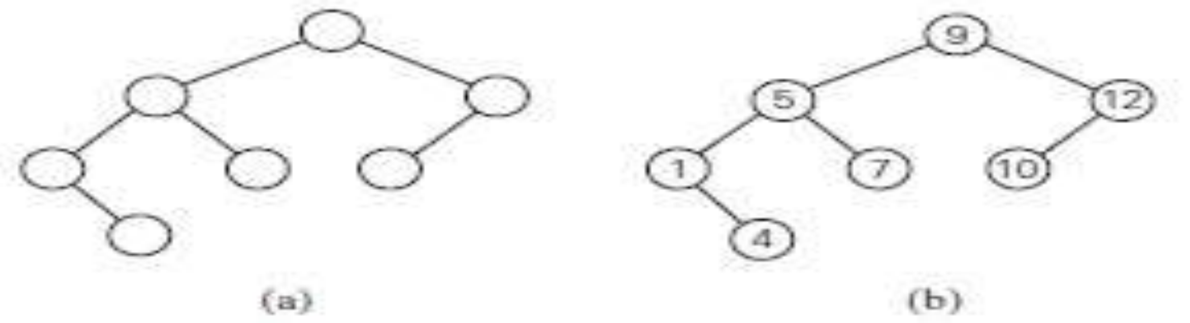


FIGURE 1.12 (a) Binary tree. (b) Binary search tree.

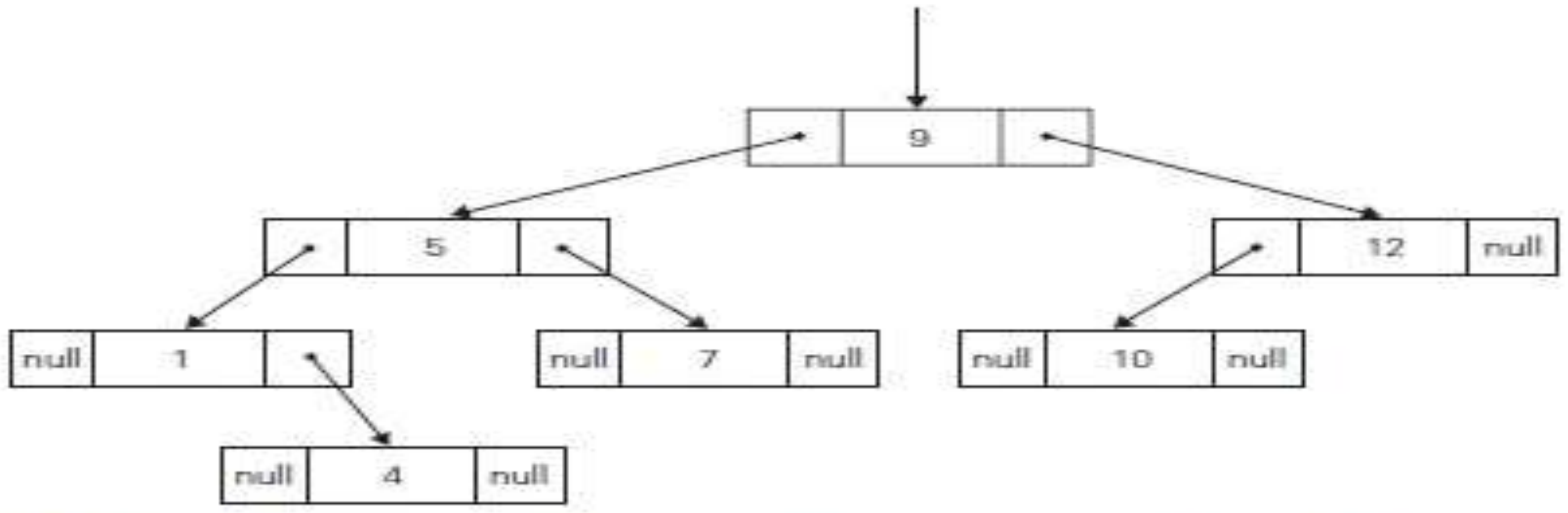


FIGURE 1.13 Standard implementation of the binary search tree in Figure 1.12b.

Trees

Binary Search Tree :

- Parental vertex is larger than all the numbers in its left subtree and smaller than all the numbers in its right subtree.

Left subtree < **Parental vertex** < **Right subtree**

Sets and dictionaries

- A **set** can be described as an unordered collection (possibly empty) of distinct items called elements of the set.

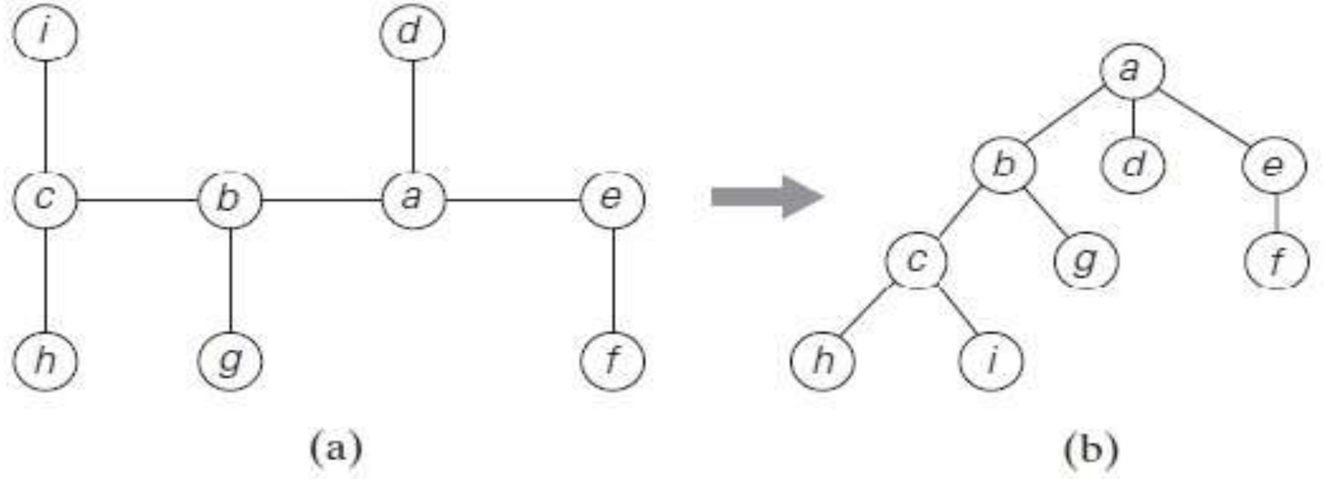
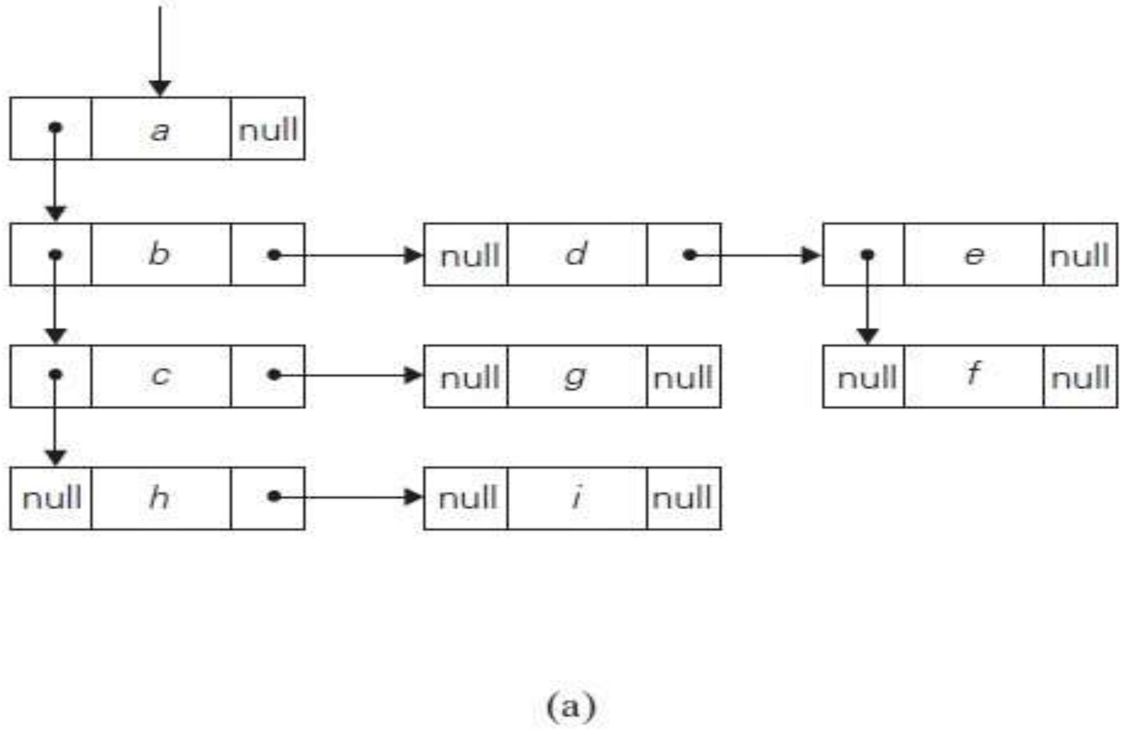


FIGURE 1.11 (a) Free tree. (b) Its transformation into a rooted tree.

Sets and dictionaries



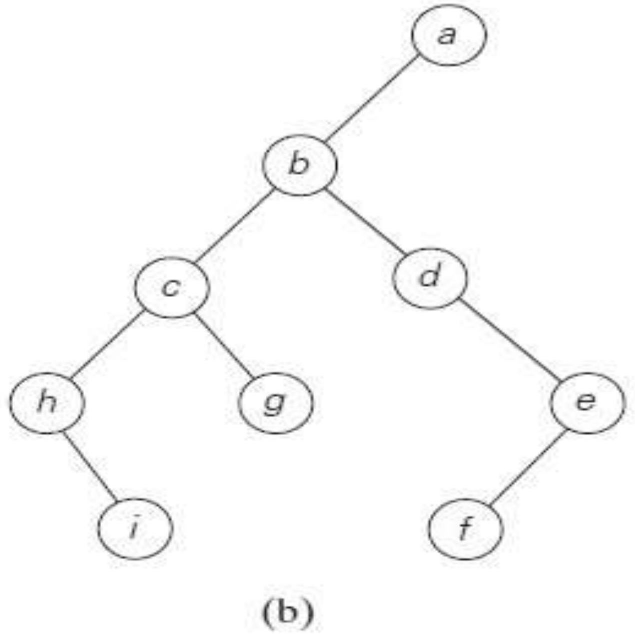
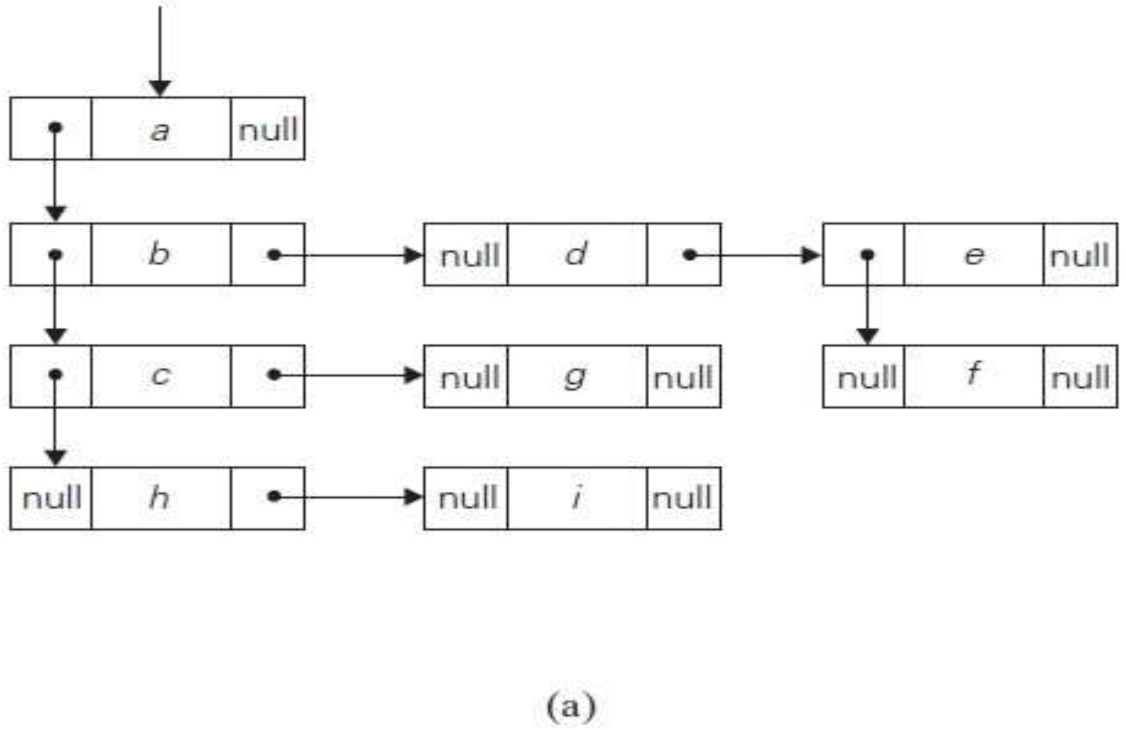


FIGURE 1.14 (a) First child–next sibling representation of the tree in Figure 1.11b. (b) Its binary tree representation.

Sets and dictionaries



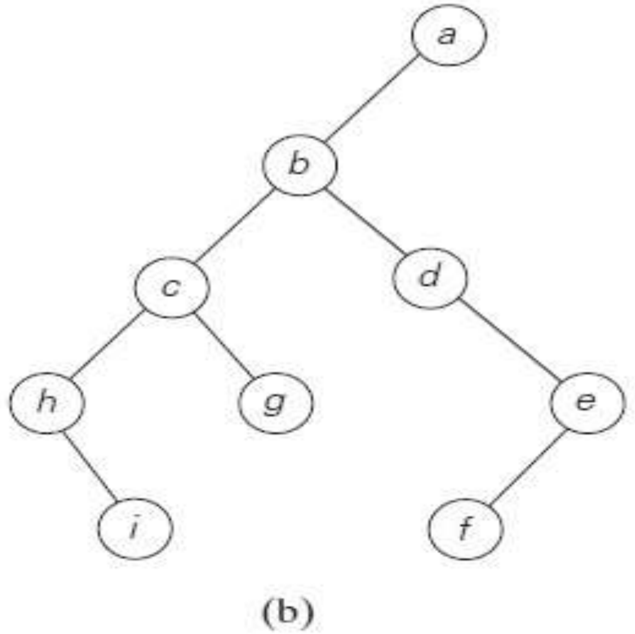


FIGURE 1.14 (a) First child–next sibling representation of the tree in Figure 1.11b. (b) Its binary tree representation.