Tree & Binary Trees

College of Computer Science, CQU

Outline

- Tree Definitions and Terminology
- Tree ADT
- Binary Tree Definitions
- Binary Tree Properties

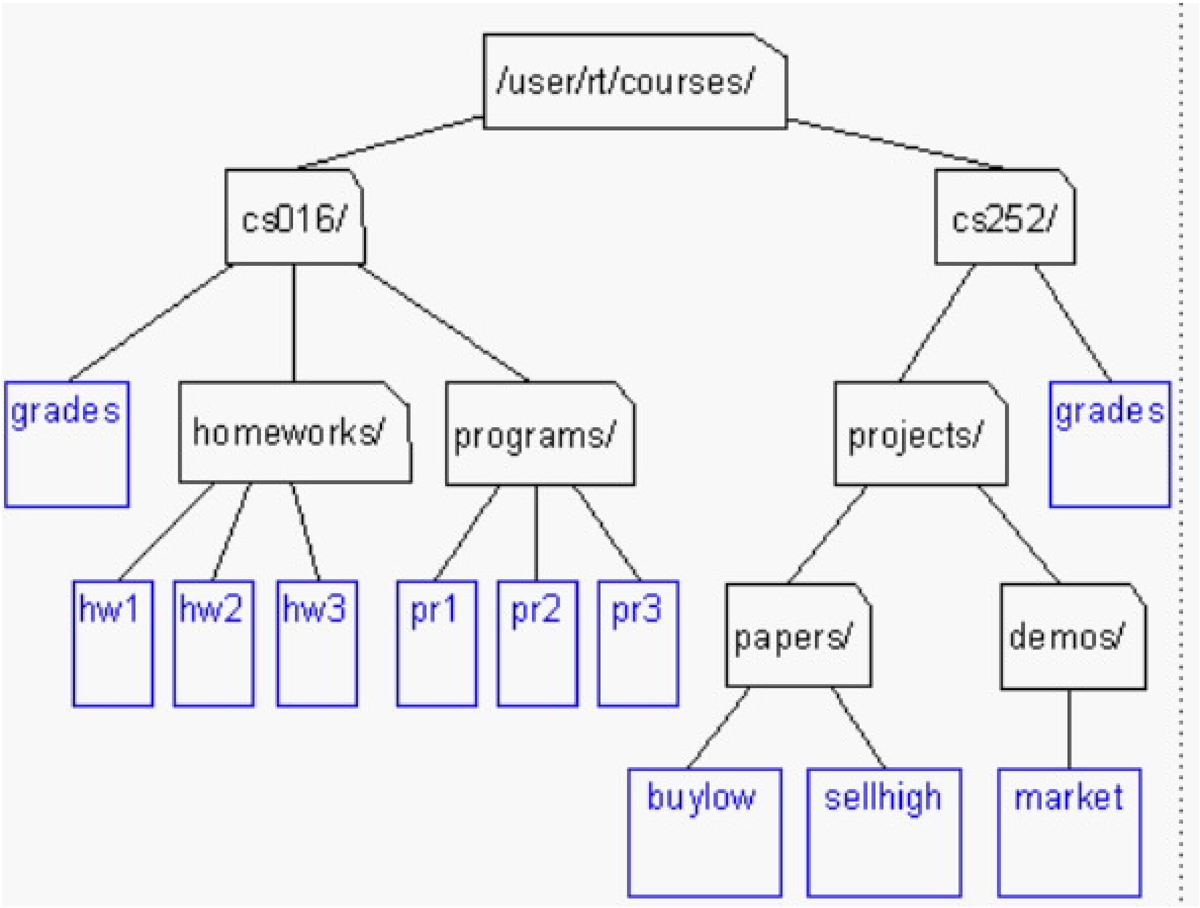
Tree Example

Representing File Structures:

- Consider the Unix file system
- Hierarchically arranged so that each file (including directories) belongs to some directory (except the / file which is the root)
- Each directory must be able to tell what files are in it

Tree Example

- Unix / Windows file structure

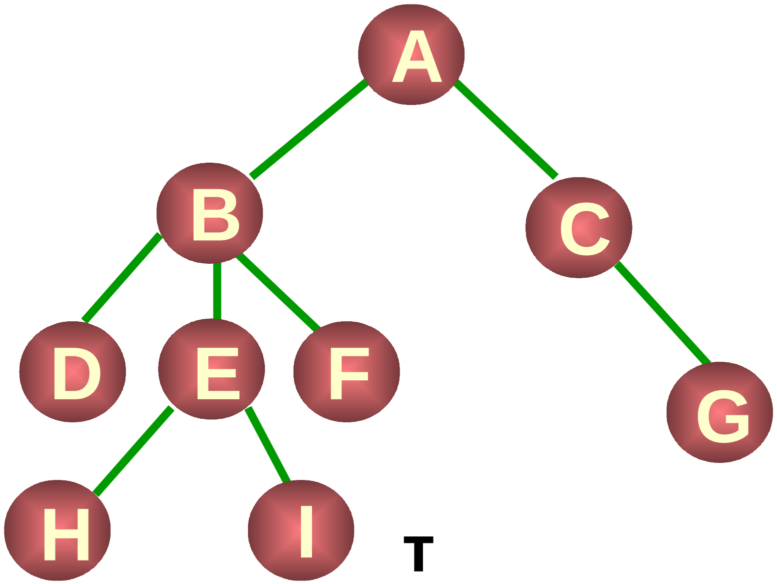


Other Trees

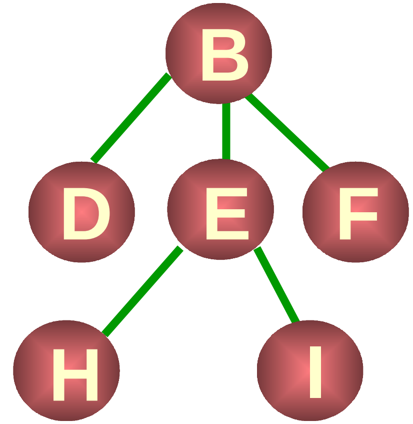
- Family Trees
- Organization Structure Charts
- Program Design
- Structure of a chapter in a book
- ……

Definition of Tree

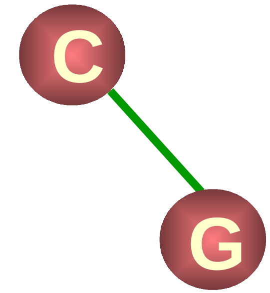
- A tree T is a finite set of one or more elements called nodes such that:
  - There is one designated node R, called the root of T.
  - If the set (T- {R}) is not empty, these nodes are partitioned into $n > 0$ disjoint subsets $T_0, T_1, \ldots, T_{n-1}$, each of which is a tree, and whose roots $R_1, R_2, \ldots, R_n$, respectively, are children of R
  - The subsets T_i (0 i < n) are said to be subtrees of T.
- root: no predecessor
- leaf: no successor
- others: only one predecessor and one or more successor
- **Definition is recursive**



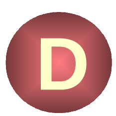
T



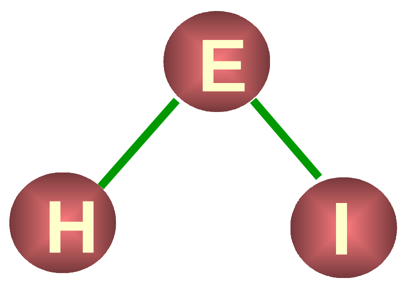
T1



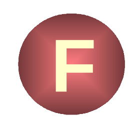
T2



T11



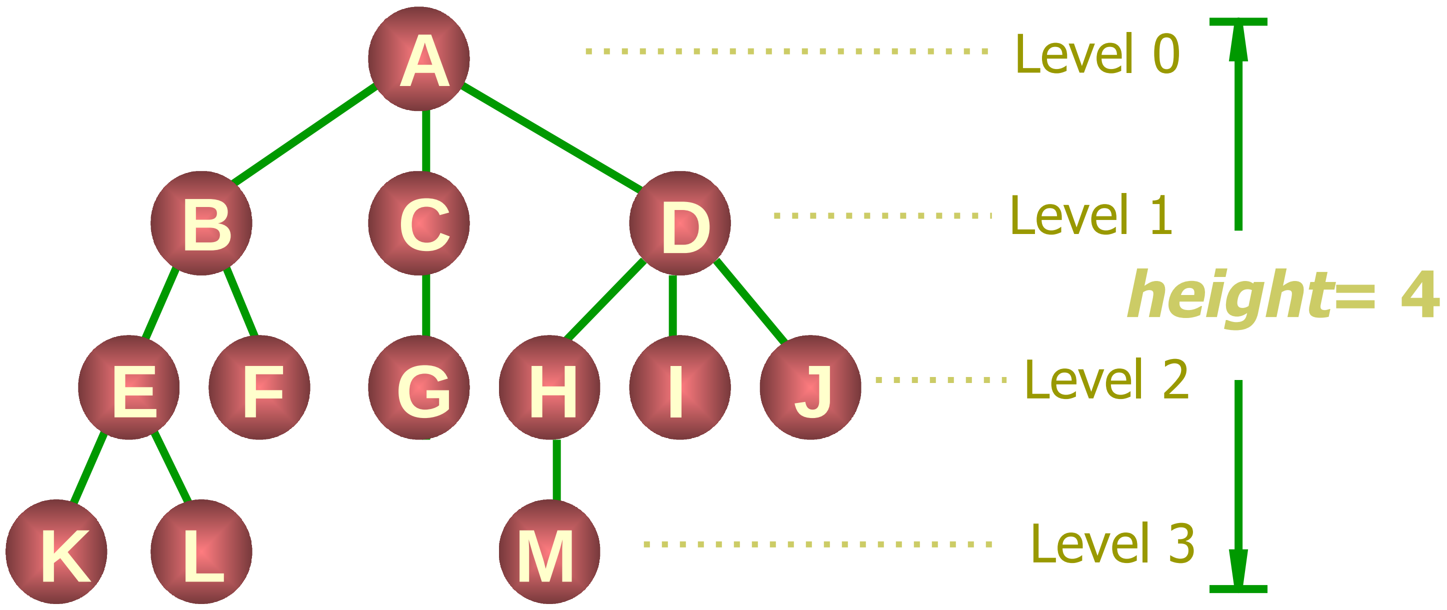
T12



T13

Tree: Level feature

- Root of subtree only have a direct previous, but can have 0 or more direct successor.



Terminology

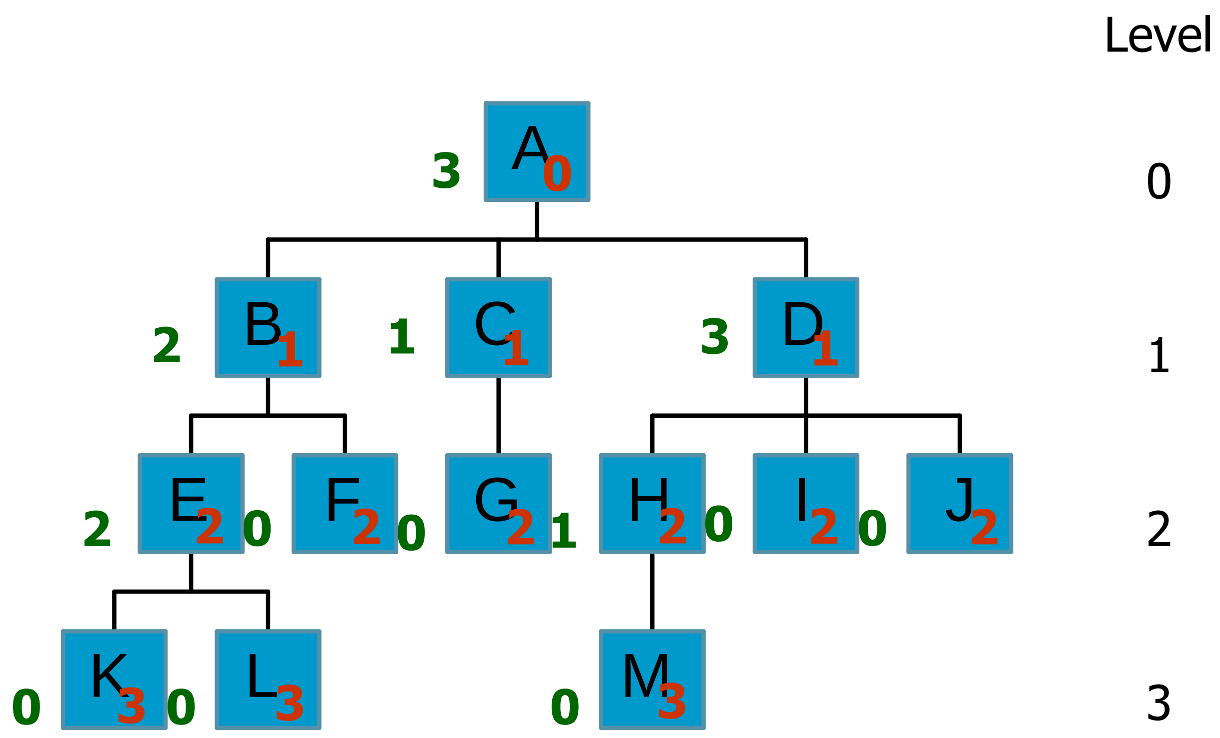
- There is an edge from a node to each of its children, and a node is said to be the parent of its children.
- If $n_1, n_2, \ldots, n_k$ is a sequence of nodes in the tree such that n_i is the parent of n_{i+1} for $1 \leq i < k$, then this sequence is called a path from n_1 to n_k. The length of the path is $k - 1$.
- The depth of a node M in the tree is the length of the path from the root of the tree to M.
- All nodes of depth d are at level d in the tree.
- The height of a tree is one more than the depth of the deepest node in the tree.

Terminology

- The degree of a node is the number of subtrees of the node
  - The degree of A is 3; the degree of C is 1.
- The degree of a tree is the maximum degree of the nodes in the tree.
- The node with degree 0 is a leaf or terminal node.
- A node that has subtrees is the parent of the roots of the subtrees.
- The roots of these subtrees are the children of the node.
- Children of the same parent are siblings.
- The ancestors of a node are all the nodes along the path from the root to the node.
- A forest is a collection of one or more trees.

Terminology

- node (13)
- degree of a node
- root
- leaf (terminal)
- internal node
- parent
- children
- sibling
- degree of a tree (3)
- ancestor
- descendant
- level of a node
- height of a tree (4)
- forest



ADT for Tree Nodes

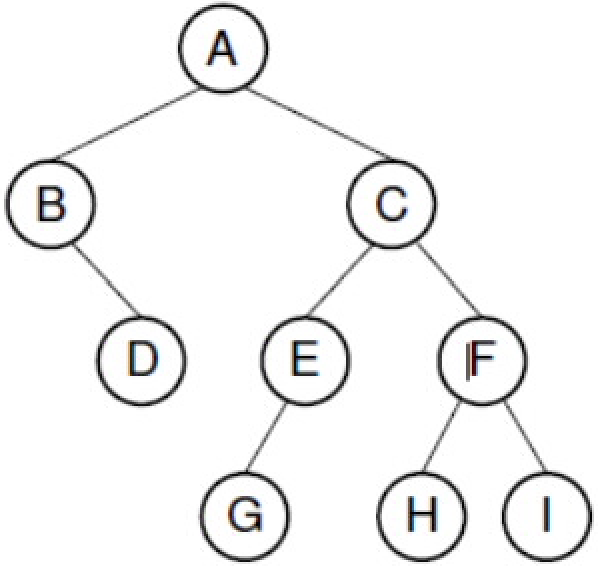
```
// General tree node ADT
template <typename  E> class GTNode {
public:
    E value(); // Return node's value
    bool isLeaf(); // True if node is a leaf
    GTNode* parent();                        // Return parent
    GTNode* leftmostChild();                 // Return first child
    GTNode* rightSibling();                  // Return right sibling
    void setValue(E&);                       // Set node's value
    void insertFirst(GTNode<E>*);            // Insert first child
    void insertNext(GTNode<E>*);             // Insert next sibling
    void removeFirst();                      // Remove first child
    void removeNext();                       // Remove right sibling
};
```

General Tree ADT

```
// General tree ADT
template <typename E> class GenTree {
public:
    void clear();                      // Send all nodes to free store
    GTNode<E>* root();                 // Return the root of the tree
    // Combine two subtrees
    void newroot(E&, GTNode<E>*, GTNode<E>*);
    void print();                      // Print a tree
};
```

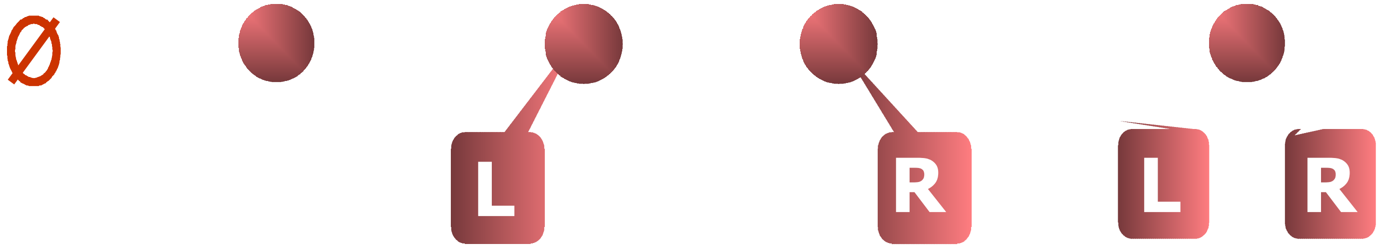
Definition of Binary Tree

- A binary tree is made up of a finite set of nodes. This set either is empty or consists of a node called the root together with two binary trees, called the left subtree and right subtree, which are disjoint from each other.
- A binary tree example

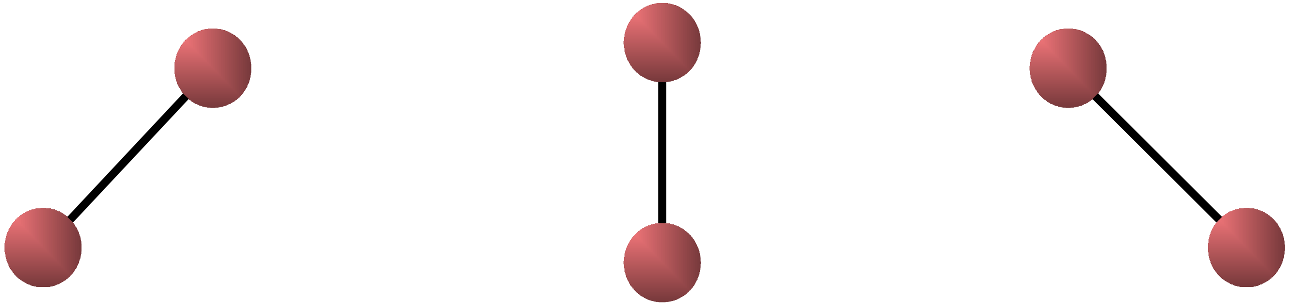


Shapes of Binary Tree

- Binary Tree has five different shapes

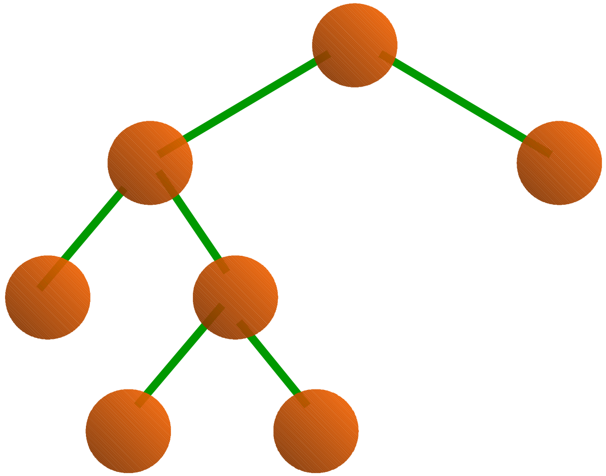


- left and right are important for binary trees
- The following three trees are the same or not?



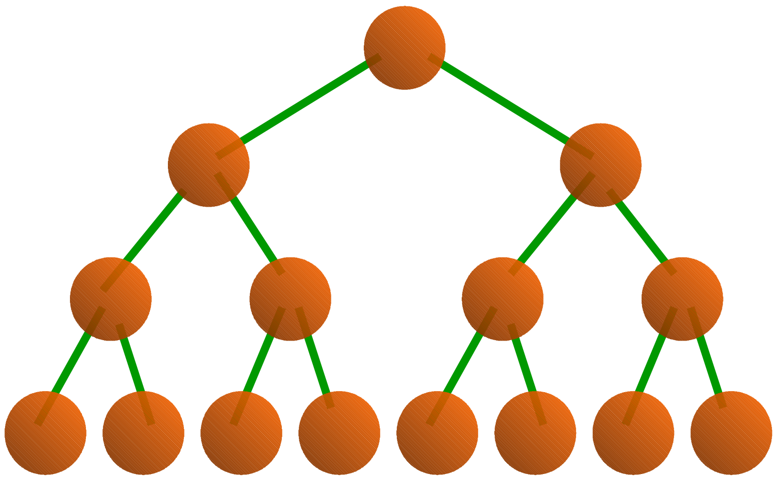
Full Binary Tree

- Each node in a full binary tree is either (1) an internal node with exactly two non-empty children or (2) a leaf.



Complete Binary Tree

- A complete binary tree has a restricted shape obtained by starting at the root and filling the tree by levels from left to right. In the complete binary tree of height d, all levels except possibly level d-1 are completely full. The bottom level has its nodes filled in from the left side.



Properties of Binary Tree

- (1) The maximum number of nodes on ith level of a binary tree is 2^{i-1}, $i \geq 1$.
- Proof: The proof is by induction on i.
  - Induction base: The root is the only node on first level. The maximum number of nodes on 1th level is $2^{i-1} = 2^0 = 1$.
  - Induction hypothesis: For all j, $1 \leq j<i$, the maximum number of nodes on jth level is 2^{j-1}.
  - Induction step: Since each node has a maximum degree of 2, the maximum number of nodes on ith level is two times the maximum number of nodes on (i–1)th level or $2 * 2^{(i-1)-1} = 2^{i-1}$

Properties of Binary Tree

- ⑵The maximum number of nodes in a binary tree of height k is 2^k-1, k≥1.
- Proof: The maximum number of nodes in a binary tree of height k is:

$$\underset{i\ \ 1}{\overset{k}{}}\ \text{(max number of nodes on ith level)}$$

$$\underset{i\ \ 1}{\overset{k}{}}\ 2^{i\ \ 1}\quad 2^{k}\quad 1$$

Relations between Number of Leaf Nodes and Nodes of Degree 2

- ⑶ For any nonempty binary tree, T, if n_0 is the number of leaf nodes and n_2 is the number of nodes of degree 2, then $n_0=n_2+1$
- proof:
  - Let n and B denote the total number of nodes & branches in T.
  - Let n_0, n_1, n_2 represent the nodes with no children, single child, and two children respectively.
  - $n= n_0+n_1+n_2$, $B+1=n$, $B=n_1+2n_2$ ==> $n_1+2n_2+1= n$, $n_1+2n_2+1= n_0+n_1+n_2$ ==> $n_0=n_2+1$

Complete Binary Tree

- ⑷The height k of the complete binary tree with n nodes is:

  $k = \log_2 n + 1$

- Proof:

  By (2) and definition of complete BT, there is:

  $2^{k-1}-1 < n \le 2^k-1$

  $2^{k-1} \le n < 2^k$

  $k-1 \le \log_2 n < k$

  k is integer, so $k = \log_2 n + 1$

Binary Tree Representations

- ⑸ If a complete binary tree with n nodes (height $=\log_2 n + 1$) is represented sequentially, then for any node with index i, $1 \le i \le n$, we have:
  - parent(i) is at i/2 if i!=1. If i=1, i is at the root and has no parent.
  - leftChild(i) is at 2i if $2i \le n$. If 2i>n, then i has no left child.
  - rightChild(i) is at 2i+1 if $2i + 1 \le n$. If 2i +1 >n, then i has no right child.
- **Proof**

Full Binary Tree Theorem

- **Theorem 5.1 Full Binary Tree Theorem:** The number of leaves in a non-empty full binary tree is one more than the number of internal nodes.
- **Proof:**

  By definition, all internal nodes of full binary tree are nodes of degree 2.

  And $n_0=n_2+1$

Binary Tree Theorem

- **Theorem 5.2** The number of empty subtrees in a non-empty binary tree is one more than the number of nodes in the tree.
- **Proof :**
  - Let n denotes the number of nodes in binary tree T.
  - By definition, every node in binary tree T has two children. So, there are 2n children in T.
  - Every node except the root node has one parent, for a total of n-1 nodes with parents. In other words, there are n-1 non-empty children.
  - The number of empty children is :$2n-(n-1)=n+1$

Reference

- Chapter 5
  - 5.1: P151--P155

- 《数据结构（ C 语言版）》，严蔚敏，吴伟民编著，清华大学出版社，1997年第1版，P118-125

-End-