

Lesson06——树形结构

【本节目标】

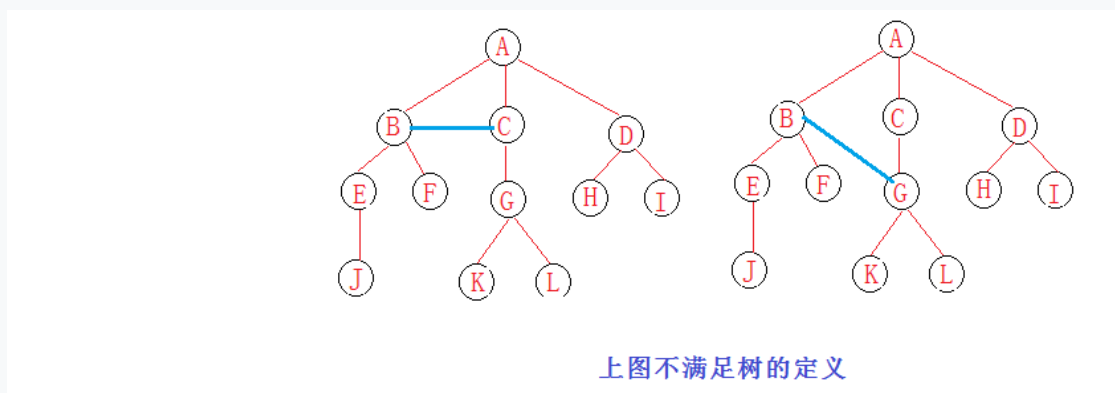
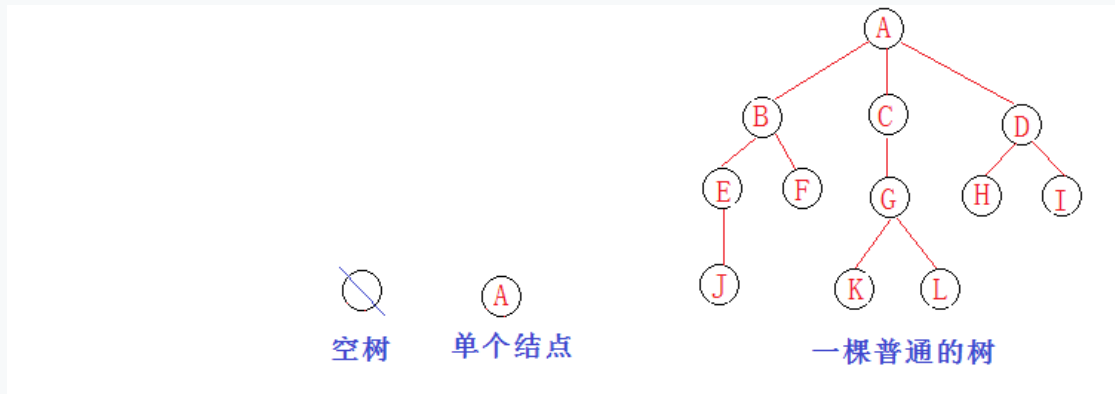
- 树的基本概念
- 树的表示方法
- 树的存储形式

树的概念

树：由 $N(N \geq 0)$ 个结点构成的集合。对 $N > 1$ 的树，有：

- 有一个特殊的结点，称为根结点，根节点没有前驱结点
- 除根节点外，其余结点被分成 $M(M > 0)$ 个互不相交的集合 T_1 、 T_2 、……、 T_m ，其中每一个集合 $T_i(1 \leq i \leq m)$ 又是一棵结构与树类似的子树。每棵子树的根结点有且只有一个前驱，可以有0个或多个后继

因此，树是递归定义的。



结点：结点包括一个数据元素及若干指向其他子树的分支(指针(索引))

结点的度：结点所拥有子树的个数称为该结点的度

叶结点：度为0的结点称为叶结点，叶节点也称为终端节点

分支结点：度不为0的结点称为分支结点，分支结点也称为非终端节点。一棵树中除叶节点外的所有节点都是分支结点

祖先结点：从根节点到该结点所经分支上的所有节点

子孙结点：以某节点为根节点的子树中所有节点

双亲结点：树中某节点有孩子结点，则这个结点称为它孩子结点的双亲结点，双亲结点也称为前驱结点

孩子结点：树中一个节点的子树的根节点称为该结点的孩子结点，孩子结点也称为后继结点

兄弟结点：具有相同双亲结点的结点称为兄弟结点

树的度：树中所有节点的度的最大值称为该树的度

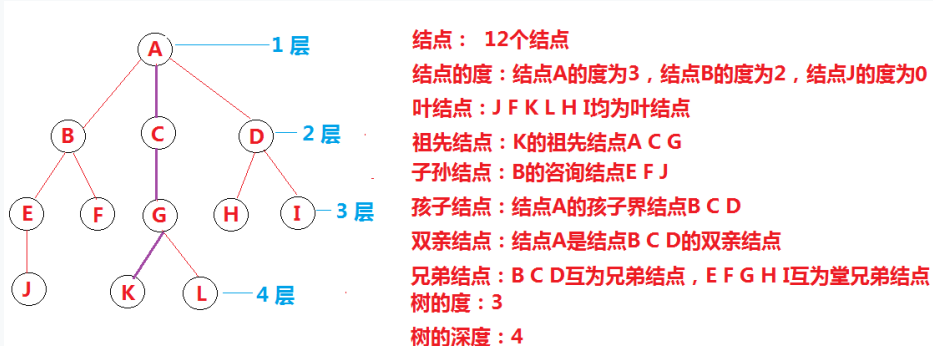
结点的层次：从根节点到树中某节点所经路径上的分支数称为该结点的层次，根节点的层次为1，其他结点层次是其双亲结点层次加1

树的深度：树中所有节点的层次的最大值称为该树的深度

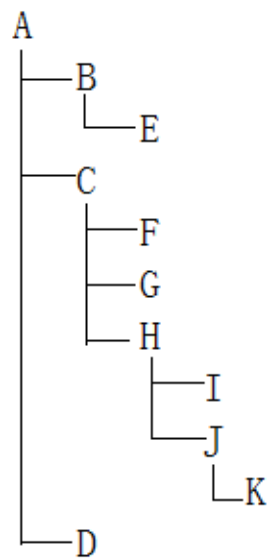
有序树：树中结点的各棵子树 T_0 、 T_1 ...是有序的，即为有序树。其中 T_1 叫做根的第一棵子树， T_2 叫着根的第二棵子树

无序树：树中结点的各棵子树之间的次序不重要，可以相互交换位置

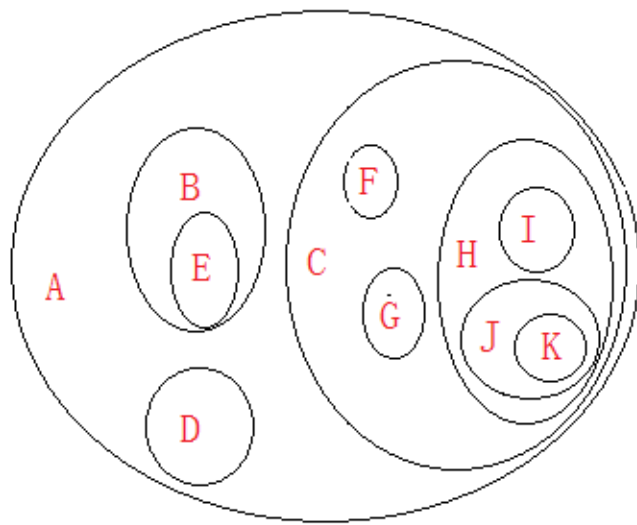
森林：树 m 棵树的集合(m 大于等于0)。在自然界中树和森林是两个不同的概念，但在数据结构中，它们之间的差别很小。删去一棵非空树的根节点，树就变成森林；反之若增加一个节点，让森林中每一棵树的根节点都变成他的子女，森林就变成一棵树



树的表示方法



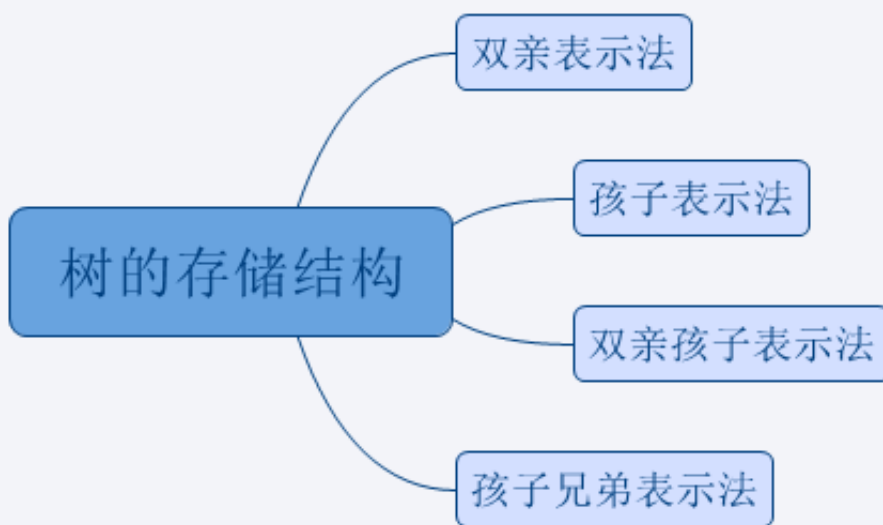
目录结构表示



集合文氏图表示

树的存储结构

计算机中存储树的信息，要求既要存储结点的数据元素信息，又要存储结点之间的逻辑关系信息。

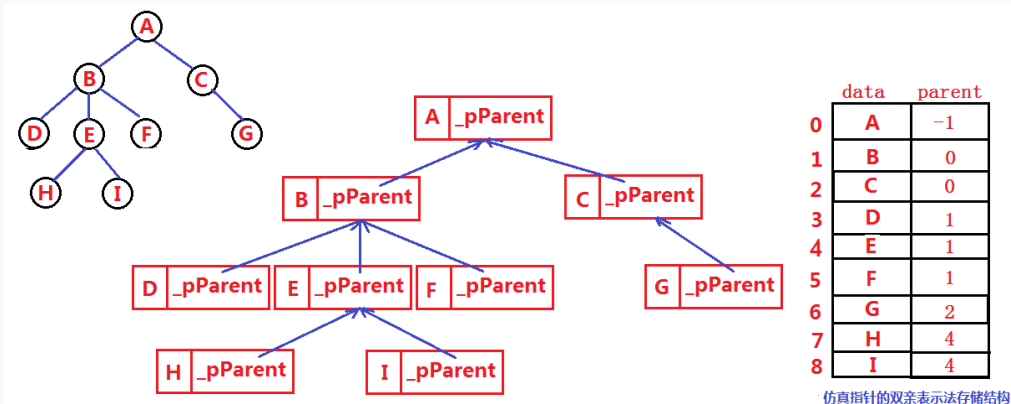


- 双亲表示法：用指针表示出每个结点的双亲结点

```

1. //树中结点类型可定义如下
2. typedef int DataType;
3. struct Node
4. {
5.     struct Node* _pParent; // 指向双亲结点指针域
6.     DataType _data;        // 结点中的数据
7. };

```



优点：寻找一个节点的双亲结点操作实现很方便

缺点：寻找一个节点的孩子结点很不方便

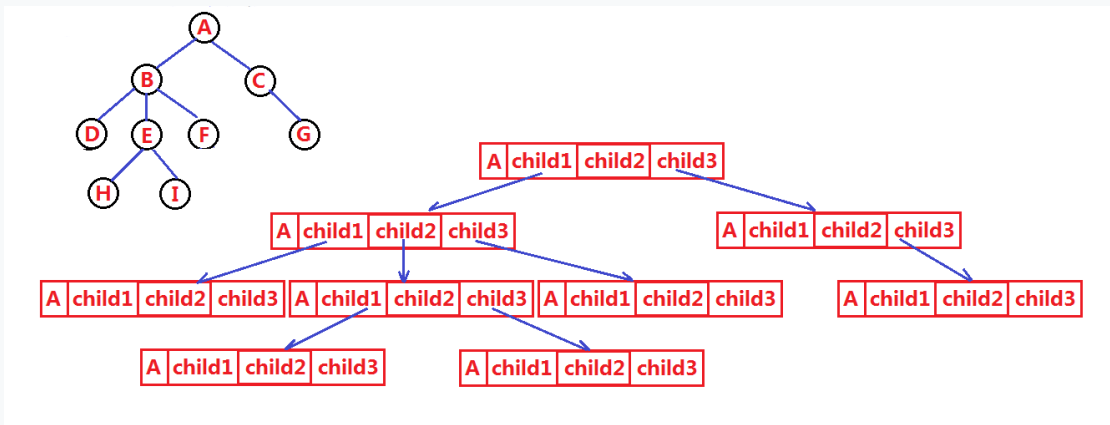
- **孩子表示法：**用指针指出每个节点的孩子结点

问题：结点的数据结构中应该给几个字段呢？

```

1. typedef int DataType;
2. struct Node
3. {
4.     struct Node* _pChild1;
5.     struct Node* _pChild2;
6.     struct Node* _pChild3;
7.     DataType _data
8. };

```



优点：寻找一个节点的孩子节点比较方便

缺点：寻找一个节点的双亲节点很不方便

- **双亲孩子表示法：**用指针既表示出每个节点的双亲节点，也表示出每个节点的孩子节点，即：双亲表示法+孩子表示法

```

1. typedef int DataType;
2. struct Node
3. {
4.     struct Node* _pParent;
5.     struct Node* _pChild1;
6.     struct Node* _pChild2;
7.     struct Node* _pChild3;
8.     DataType _data
9. };

```

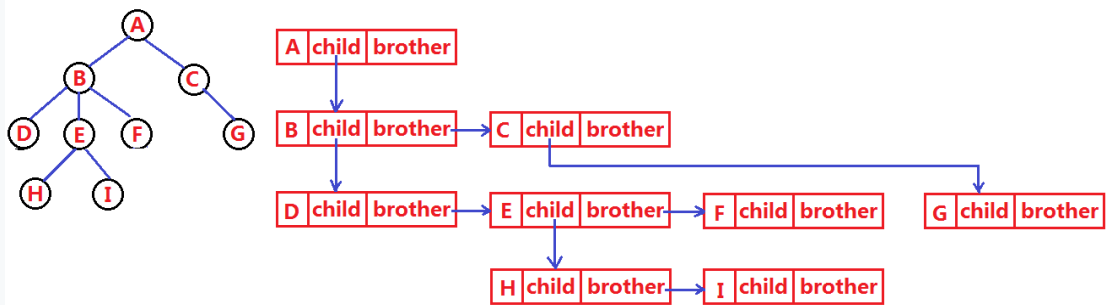
优点：找某个节点的双亲节点和孩子节点非常方便

- **孩子兄弟表示法：**即表示出每个节点的第一个孩子节点，也表示出每个节点的下一个兄弟节点

```

1. typedef int DataType;
2. struct Node
3. {
4.     struct Node* _pChild1;           // 第一个孩子结点
5.     struct Node* _pNextBrother;      // 指向其下一个兄弟结点
6.     DataType _data;                  // 结点中的数据域
7. };

```



树的应用

计算机

Windows7_OS (C:)

新加卷 (D:)

360安全浏览器下载

ApplySoft

Game

Program Files (x86)

ProgramSoft

Rose

SourceInside

V6.0

VS CSDN 精简版

vs2010

VS2010助手

vs2013

VS2013助手

印象笔记

新加卷 (E:)

Working

WorkStadio