

第二十一章 树

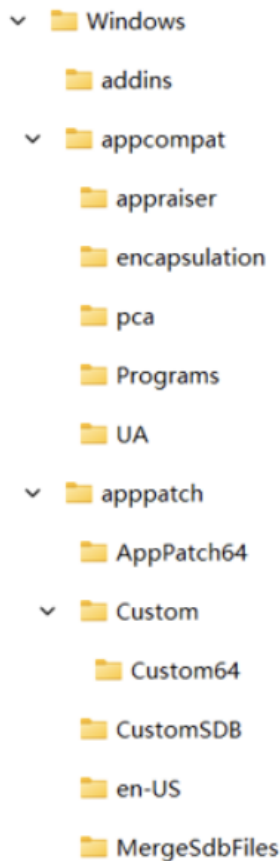
1. 树的定义

我们认识的大自然的树一般主要由根、干、枝、叶组成。

树枝从树干开始分叉，到最后是叶子。



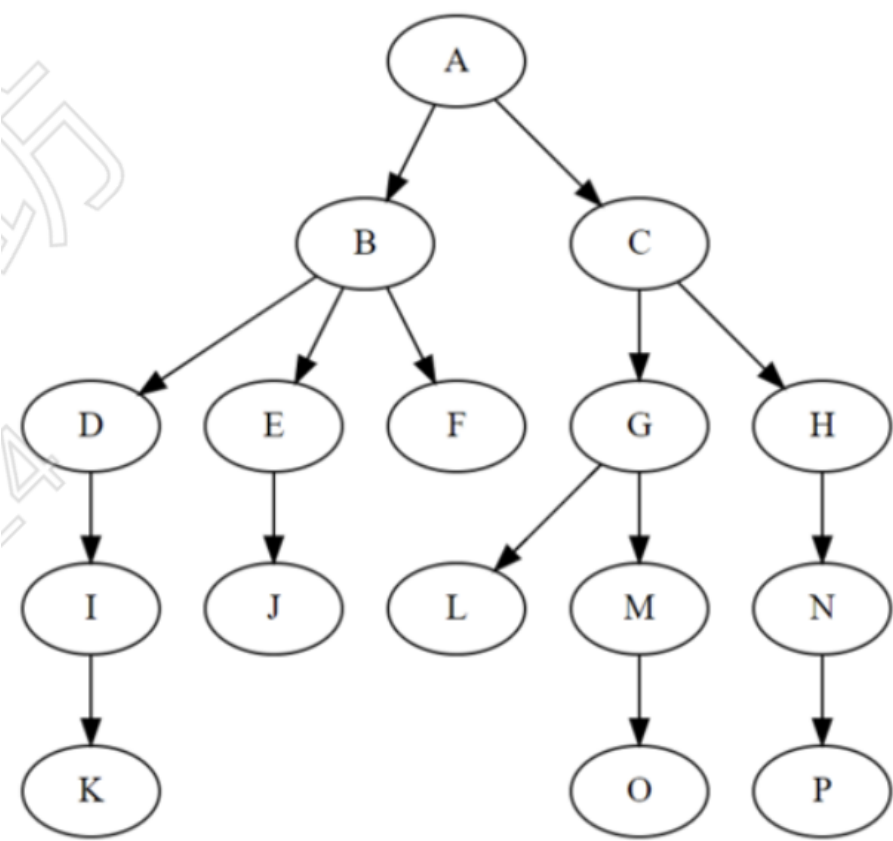
- 树状结构
很多概念都有类似树这样的分层分支的结构，例如： 组织结构
目录结构



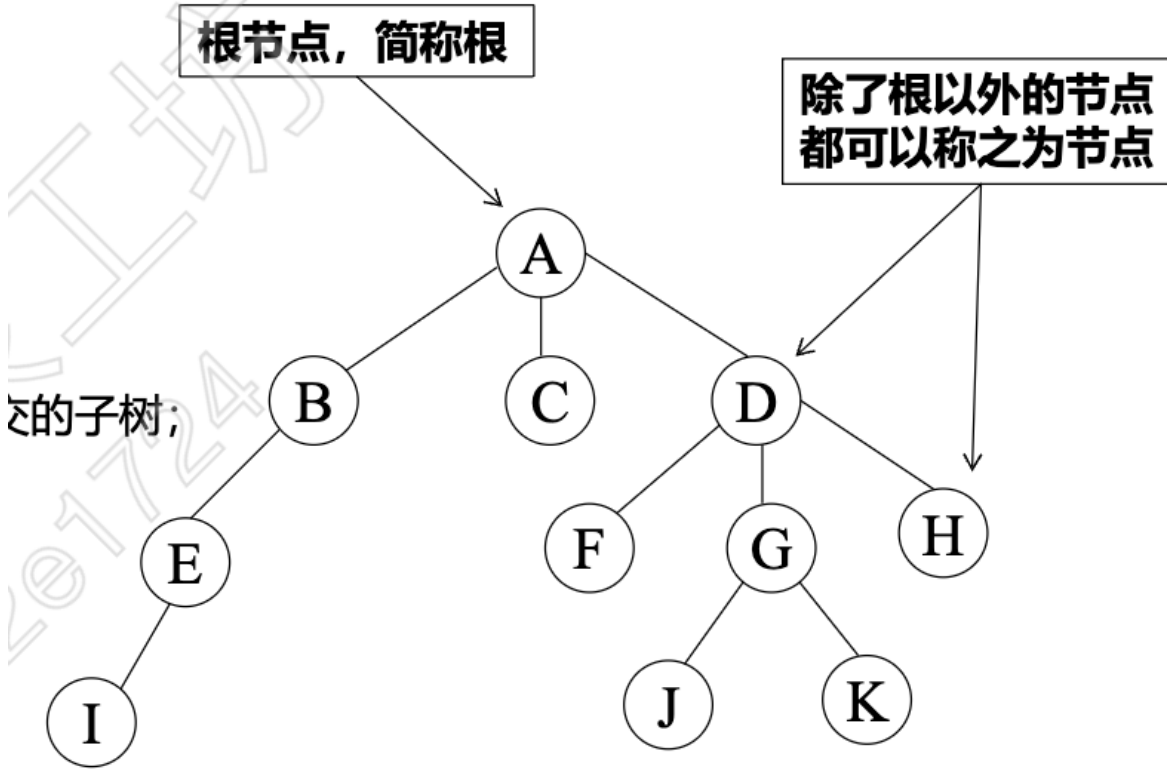
在计算机科学中，树（英语：tree）是一种抽象数据类型，或是实现这种抽象数据类型的数据结构，其主要用来模拟具有树状结构性质的数据集合

它是由 n ($n > 0$) 个有限节点（结点）组成一个具有层次关系的集合。把它叫做“树”是因为它看起来像一棵倒挂的树，也就是说它是根在上面，而叶在下面的。

树是一个一对多的结构，这是一种非线性结构。（树中的一个节点有一个父节点（除根节点外）和多个子节点，而线性数据结构中一个节点最多只有一个直接前驱和一个直接后继）

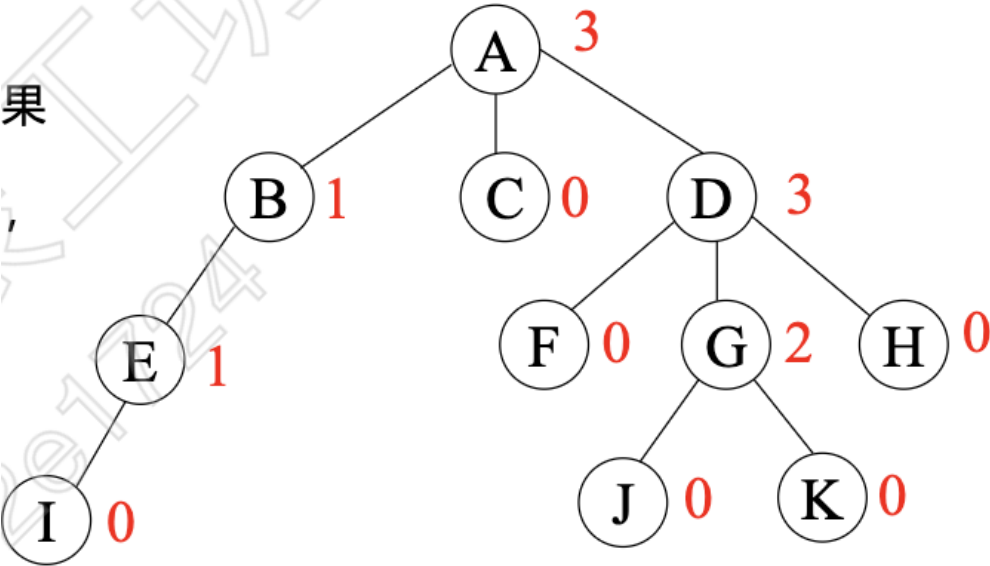


- 树的特点
 - 每个节点都只有有限个子节点或无子节点
 - 没有父节点的节点称为根节点
 - 每一个非根节点有且只有一个父节点
 - 除了根节点外，每个子节点可以分为多个不相交的子树；树里面没有环路（cycle）



2. 树的基本概念

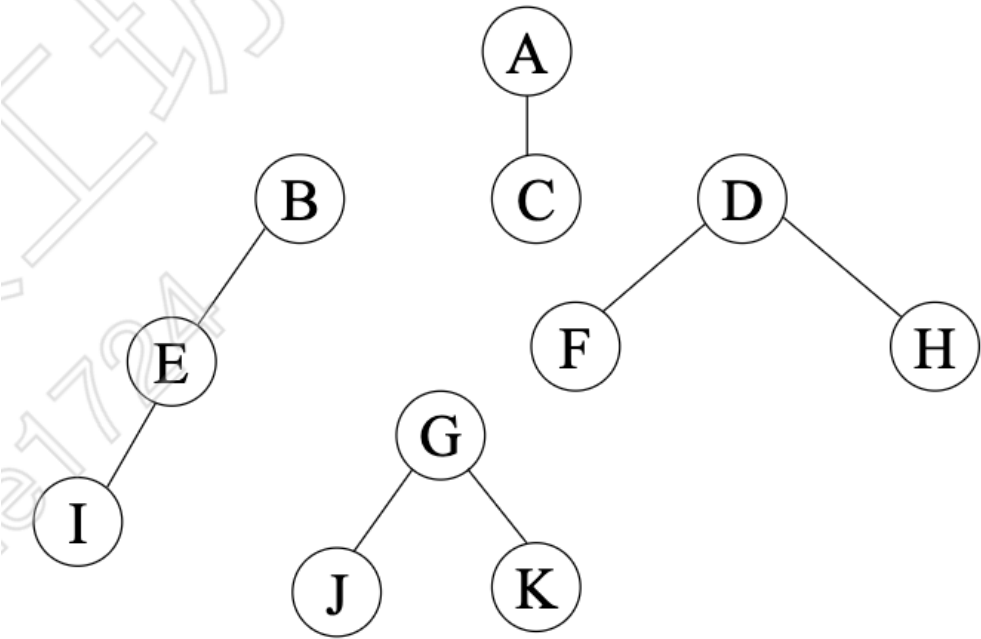
节点与边：
在树这种数据结构中，每一个元素被称为“节点”，或结点”。每个节点可以存储数据（例如一个数值或者一个字符串），并且可以与其他节点相连
图中的每个圆圈都代表一个节点
连接节点的线称为“边”，节点都通过边相连



在树中采用边连

接两个相关联的节点，如D和G。
其中D是上端节点，G是下端节点。称D是的父节点，G为D的子节点
同时，G又是j的父节点
除了最顶层的根节点外，每个节点都只有一个父节点，并且有零个或多个子节点
根节点： 没有父节点的节点称为根节点
节点的祖先： 从根节点到该节点所有的节点，包含该节点本身，都是该节点的祖先ADGJ都是J的祖先
节点的子孙： 某节点为根的子树任意节点包含该节点的本身都是该节点的子孙，途中ADGJ都是A的子孙。
兄弟节点： 同一父节点的不同子节点，互称为兄弟节点，如FGH统称为兄弟节点
节点的度： 一个节点的子节点的数量叫做这个节点的度，如果一个节点没有子节点那么他的度为0，如果该节点有一个子节点他的度为1，如果有两个子节点，他的度为2。
树的度： 树的度是树中节点的最大分支数，例如图中该树的度为3
叶节点： 度为0的节点称为叶节点或者叶子节点，终端节点
终端节点： 度不为0的节点为分支节点（或非终端节点）
节点的层次： 从根节点开始，根节点为第一层，根节点的子节点为第二层，图中G为第三层
节点的深度： 节点的深度就是从根节点到接待你的路径长度。图中G的深度为2
树的深度： 数的深度就是一棵树中节点的最大深度就是树的深度。这颗树的深度为3

森林：由m棵互不相交的树的集合就是森林



3.树的存储结构

树的存储结构包括以下四种：

- 父节点表示法
- 子节点表示法
- 父子节点表示法
- 孩子兄弟节点表示法

父节点表示法

树中每个节点可能有多个子节点，但是一个节点最多有一个父节点，父节点表示法用顺序表来存储，每个节点只需要增加一个父节点域。

```
struct TreeNode{
    int data;
    int paratIndex;
} nodes[100];
```

示例：

例如：F、G、H 节点的父节点是 D，D 节点的索引是 3，所以它们的 parentIndex 是 3

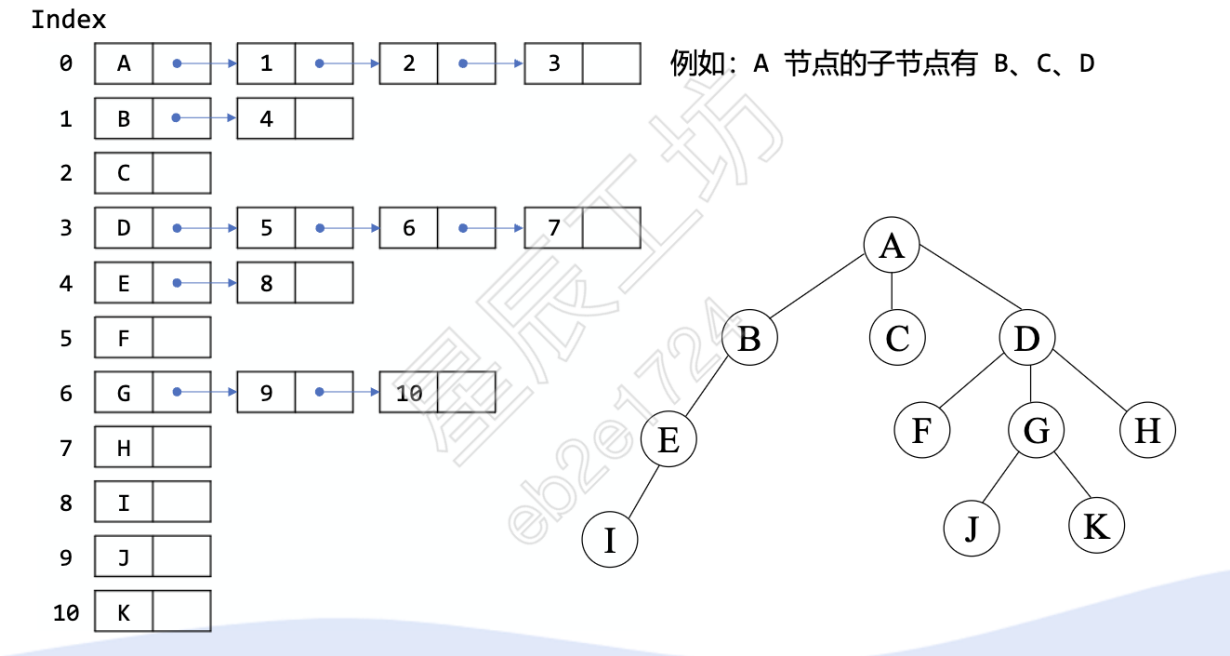
Index	0	1	2	3	4	5	6	7	8	9	10
data	A	B	C	D	E	F	G	H	I	J	K
parentIndex	-1	0	0	0	1	3	3	3	4	6	6

子节点表示法

子节点表示法普遍采用的是顺序表加链表的方式进行存储，节点使用链表保存所有的子节点

```
struct TreeNode{
    int data;
    vector<int> children;
};
vector<TreeNode> tree;
```

示例：



父子节点表示法

还可以将两者结合起来，这样节点捅死具有父节点和子节点的信息

```
struct TreeNode{
    int data;
    int parentIndex;
    vector<int> children;
};
vector<TreeNode> tree;
```

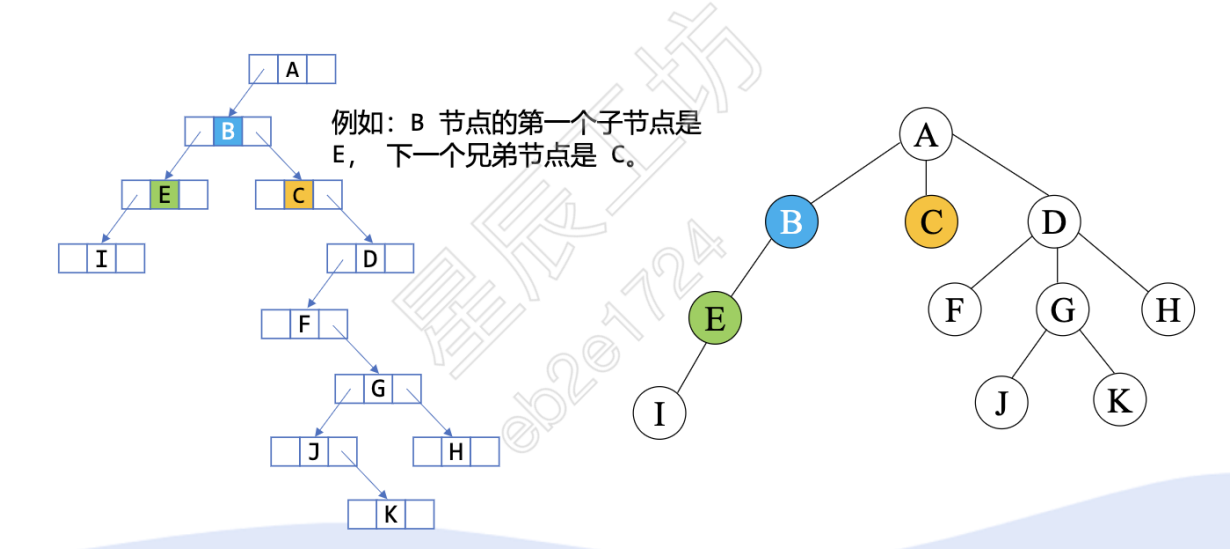
孩子兄弟节点表示法

也叫做二叉树形表示法，每个节点除了指向自己的第一个孩子节点外，还指向下一个兄弟节点。

```
struct TreeNode{
    int data;
    int firstChild; //第一个子节点
    int nextSibling; //下一个兄弟节点
};
vector<TreeNode> tree;
```

左边指针指向第一个孩子，右边指针指向自己的兄弟

从树的根节点开始，依次用链表存储各个节点的第一个孩子节点和其下一个兄弟节点。



4. 树的遍历

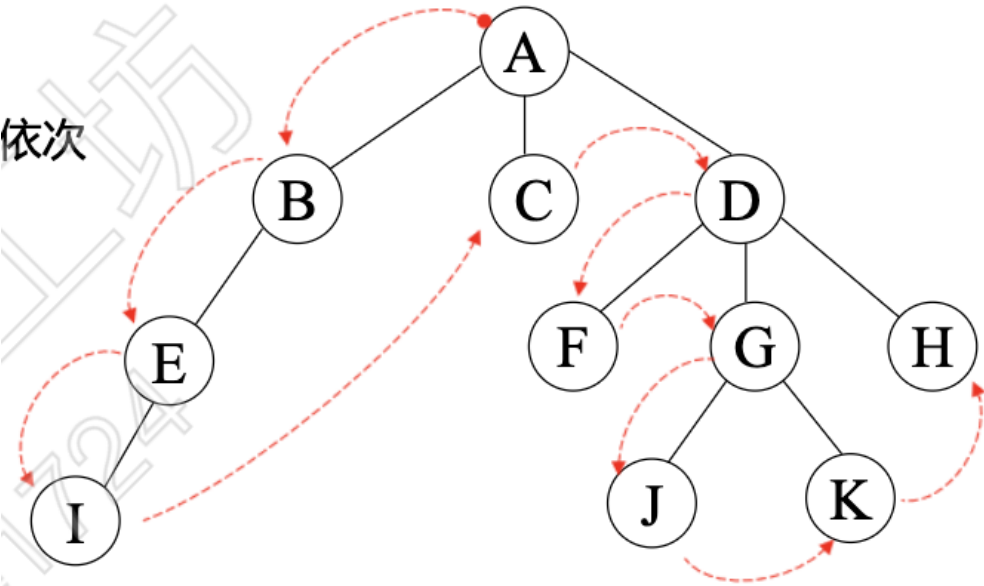
树的遍历按照一定的规律不重复的访问树的每一个节点，遍历就是将非线形的树状结构按照一定的规律转为非线形结构

深度优先遍历：先序遍历和后序遍历

广度优先遍历：层次遍历

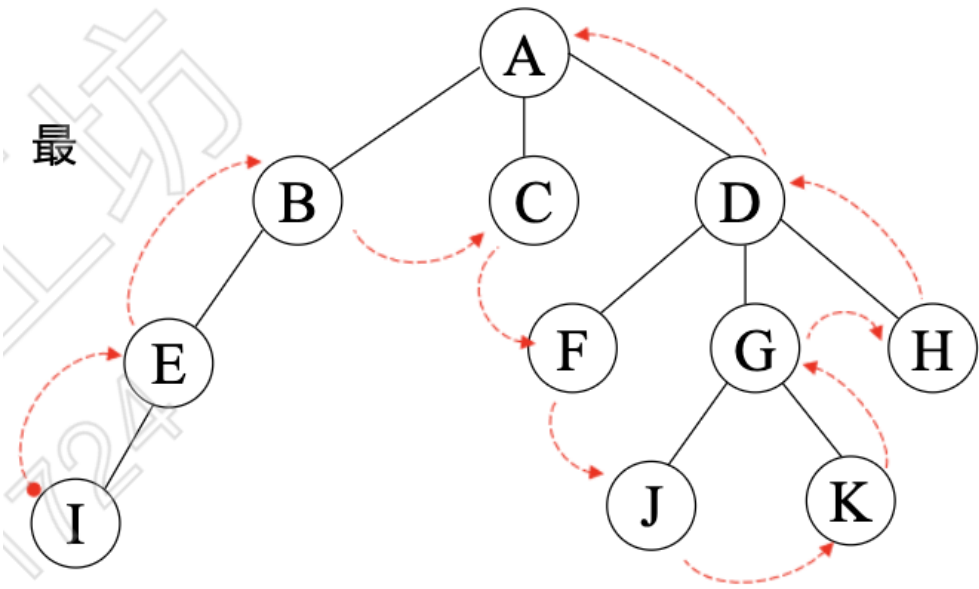
先序遍历

先访问根节点，然后根据先序遍历的原则从左到右依次访问每一个子树

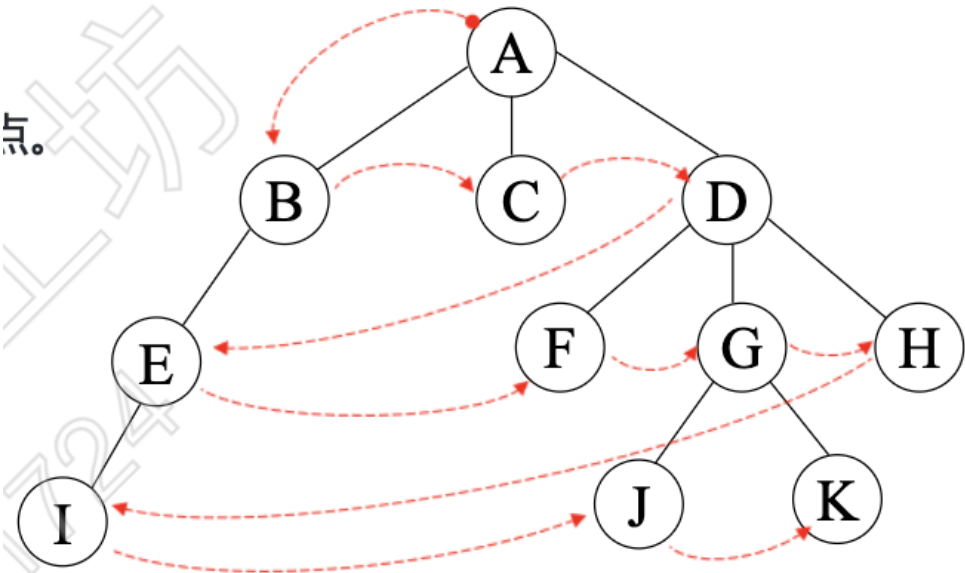


先序遍历的结果是ABEICDFGJKH

后序遍历 按照后序遍历的原则从左到右依次访问每一棵子树，最后访问根节点。



层次遍历
按照从上到下，从左到右的顺序依次访问每个节点



ABCDEFGHIJK

层次遍历的结果：

5.树的其他表示方式（了解即可）

除了使用树形结构图表示树之外，还有其它方法
嵌套集合表示法 凹入表示法 管仪表表示法