

2020/06/04_数据结构_第5课_树、二叉查找树

笔记本： 数据结构

创建时间： 2020/6/4 星期四 15:40

作者： ileemi

标签： 二叉查找树, 树

- [树](#)
 - [树的结点](#)
 - [子树和空树](#)
 - [结点的高度](#)
 - [结点的深度](#)
 - [结点的层](#)
 - [有序树和无序树](#)
 - [总结](#)
- [二叉树](#)
 - [满二叉树](#)
 - [二叉树的性质](#)
 - [完全二叉树](#)
- [二叉查找树（二叉搜索树、二叉排序树）](#)
 - [二叉查找树](#)
 - [二叉查找树的性质](#)

树

树结构是一种非线性存储结构，存储的是具有“一对多”关系的数据元素的集合。

树的结点

结点：使用树结构存储的每一个数据元素都被称为结点。

树中相连的结点具有**父子关系**，每个结点只有一个父节点，多个子节点。

树根结点（根节点）：没有父结点的结点称之为**根节点**。每一个非空树都有且只有一个被称为根的结点。

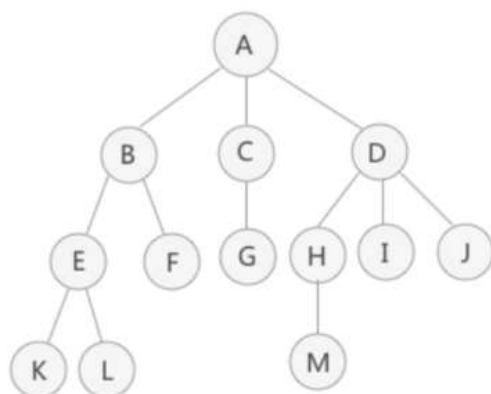
树根的判断依据为：如果一个结点没有父结点，那么这个结点就是整棵树的根结点。

叶子结点：如果结点没有任何子结点，那么此结点称为叶子结点（叶结点）。

分支节点：有父亲有儿子的节点称为分支节点。

子树和空树

子树：下图中，整棵树的根结点为结点 A，而如果单看结点 B、E、F、K、L 组成的部分来说，也是棵树，而且节点 B 为这棵树的根结点。所以称 B、E、F、K、L 这几个结点组成的树为整棵树的子树；同样，结点 E、K、L 构成的也是一棵子树，根结点为 E。



注意：单个结点也是一棵树，只不过根结点就是它本身

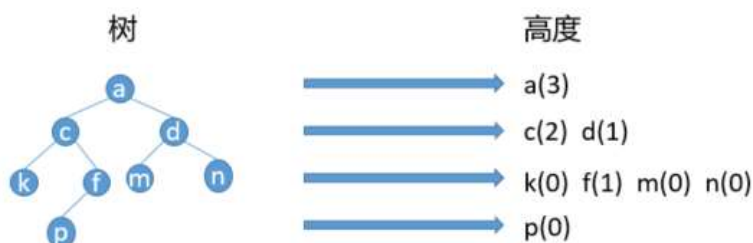
树是由根结点和若干棵子树构成的。

空树：如果集合本身为空，那么构成的树就被称为空树。空树中没有结点。

在树结构中，对于具有同一个根结点的各个子树，相互之间不能有交集。如果有交集，就破坏了树的结构，该结构就不算是一颗树。

结点的高度

结点到叶子的最长路径的边数，叶子结点的高度为0。树的高度为根结点的高度。



结点的深度

对于一个结点，拥有的子树数（结点有多少分支）称为结点的度（Degree）。

一棵树的度是树内各结点的度的最大值。

结点到根结点所经历的边数，根结点的深度为0。树的深度为深度最大的叶子结点的深度。



结点的层

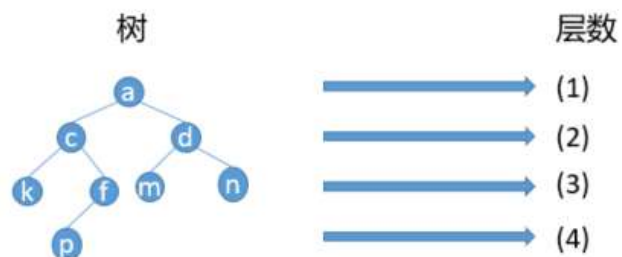
结点的层从根节点开始数，根节点为第一层，根的孩子是第二层，以此类推。

一棵树的深度（高度）是树中结点所在的最大的层次。

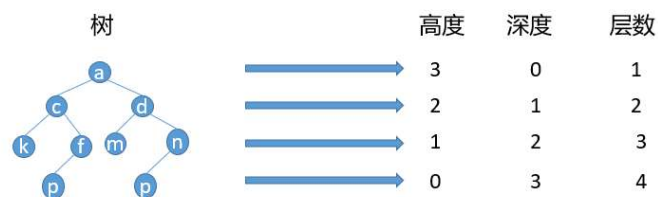
深度是距离根结点，高度是距离叶子结点

高度：从叶子往根数

深度和层数：是从根部往叶子数



• 区分 深度，高度，层



有序树和无序树

如果规定树中结点的子节点从左向右是有次序的，不能互换的，则该树为有序树，否则为无序树。

在有序树中，一个结点最左边的子树称为"第一个孩子"，最右边的称为"最后一个孩子"。

总结

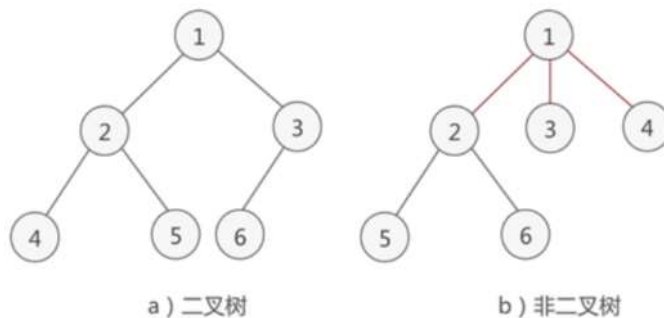
树型存储结构类似于家族的族谱，各个结点之间也同样可能具有父子、兄弟、表兄弟的关系，重点理解树的根结点和子树的定义，同时要会计算树中各个结点的度和层次，以及树的深度。

二叉树

满足以下两个条件的树就是二叉树：

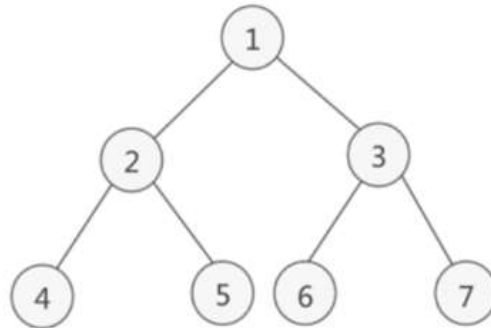
1. 本身是有序树
2. 树中包含的各个结点的度不能超过2，即只能是 0、1 或者 2

也就是说每个分支节点最多只有两个子节点，且其是有序树。



满二叉树

在一棵二叉树中，如果所有的分支结点都存在左孩子和右孩子，并且所有的叶子都在最底层，这样的树成为满二叉树。如下图，其就是一个满二叉树。

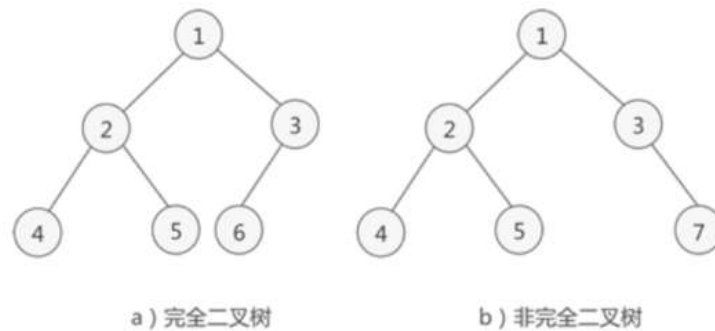


二叉树的性质

1. 在二叉树的第 n 层上最多有 $(2^{(n-1)}, n \geq 1)$ 个结点
2. 深度为 n 的二叉树最多有 $(2^n - 1, n \geq 1)$ 个结点，深度为4，总结点数为：15
3. 含有 n 个结点的满二叉树深度为 $(\log(n+1))$ ，15个结点，深度为 $\log 16 = 4$
 $2^{(n+1)} - 1 = ?$
 $2^{(x+1)} - 1 = n$
 $2^{(x+1)} = n + 1$
 $x + 1 = \log(n + 1)$
 $x = \log(n+1) - 1$

完全二叉树

如果二叉树中除去最后一层节点为满二叉树，且最后一层的结点依次从左到右分布，则此二叉树被称为完全二叉树。



上图 a) 所示是一棵完全二叉树，图 b) 由于最后一层的节点没有按照从左向右分布，所以其只能算作是一个普通的二叉树。

完全二叉树除了具有普通二叉树的性质，它自身也具有一些独特的性质，比如说， n 个结点的完全二叉树的深度为 $\lfloor \log_2 n \rfloor + 1$ 。

对于任意一个完全二叉树来说，如果将含有的结点按照层次从左到右依次标号（如上图 a)），对于任意一个结点 i ，完全二叉树还有以下几个结论成立：

1. 当 $i > 1$ 时，父亲结点为结点 $\lfloor i/2 \rfloor$ 。（ $i=1$ 时，表示的是根结点，无父亲结点）
2. 如果 $2 * i > n$ （总结点的个数），则结点 i 肯定没有左孩子（为叶子结点）；否则其左孩子是结点 $2 * i$ 。
3. 如果 $2 * i + 1 > n$ ，则结点 i 肯定没有右孩子；否则右孩子是结点 $2 * i + 1$ 。

二叉查找树（二叉搜索树、二叉排序树）

对于二叉树中的任意结点，左孩子的值小于这个结点的值，右孩子的值大于这个结点的值。在一般情况下，查询效率比链表结构要高

二叉树查找 35

查找次数最对不会超过层数

二叉树查找一个数值

n 层的二叉查找树，算法时间度为： $\log(n)$ 对数阶

二叉查找树结构比线性结构的查询速度快，二叉搜索树，查询时间复杂度为 $\log(n)$ 。

对于树中任意分支结点，左侧的结点的值都比自己小，右侧的结点值都比自己大

查询效率有要求的情况下会用到平衡树

定义：

1. 若左子树不空，则左子树上所有结点的值均小于它的根结点的值。
2. 若右子树不空，则右子树上所有结点的值均大于它的根结点的值
3. 左、右子树也分别为二叉排序树

4. 没有键值相等的结点

二叉查找树

属性：

二叉树要么是空结点的集合（空树），要么是一个根结点的结点集合

每个结点都有两个子树，分别叫做左子树和右子树

每个子树本身也是一个二叉树，也可能是一个空树

二叉查找树是一个有序的二叉树，每个结点包含一个项，左子树的所有项都在根结点的前面，右子树的所有项都在根结点的后面。

二叉查找树的性质

- 若任意节点的左子树不空，则左子树上所有结点的值均小于它的根结点的值；
- 若任意节点的右子树不空，则右子树上所有结点的值均大于它的根结点的值；
- 任意节点的左、右子树也分别为二叉查找树；
- 没有键值相等的节点。
- 对二叉查找树进行中序遍历，即可得到有序的数列

二叉查找树的查询复杂度，和二分查找一样，插入和查找的时间复杂度均为 $O(\log n)$ ，但是在最坏的情况下仍然会有 $O(n)$ 的时间复杂度。原因在于插入和删除元素的时候，树没有保持平衡。