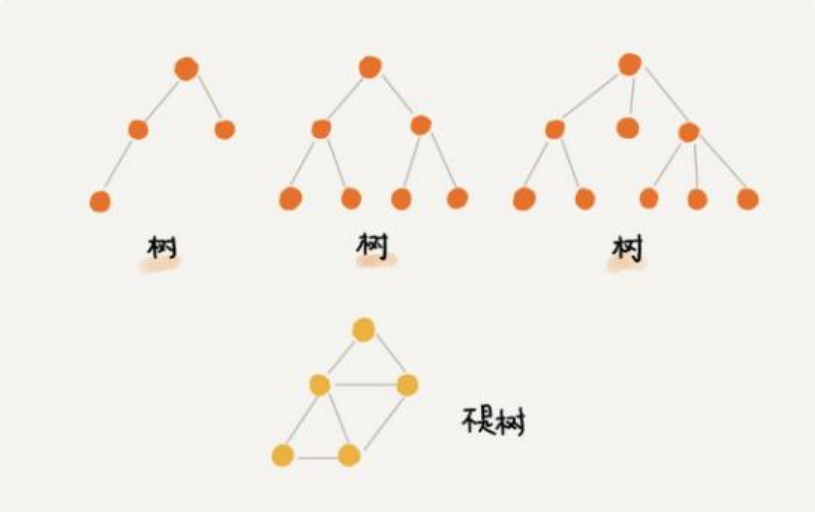
用数组存储二叉树合适么

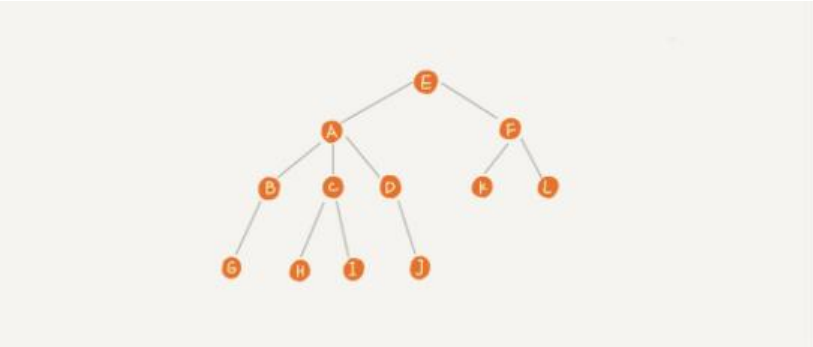
作者：zyjImmortal
原文链接：https://ld246.com/article/1577952783442
来源网站：链滴

树



里面每个元素我们叫作“节点”；用来连线相邻节点之间的关系，我们叫作“父子关系”，不存在三节点之间闭合的情况。

树的一些概念



途中A节点就是根节点，处于底部的G、H、I、J、K、L就是叶子节点，A就是B、C、D的父节点，他就是子节点，彼此之间又是兄弟节点。

节点的高度：节点到叶子节点的最长路径，也就是边数

节点的深度：根节点到这个节点所经历的边的个数

节点的层数：节点的深度+1

树的高度：根节点的高度

二叉树

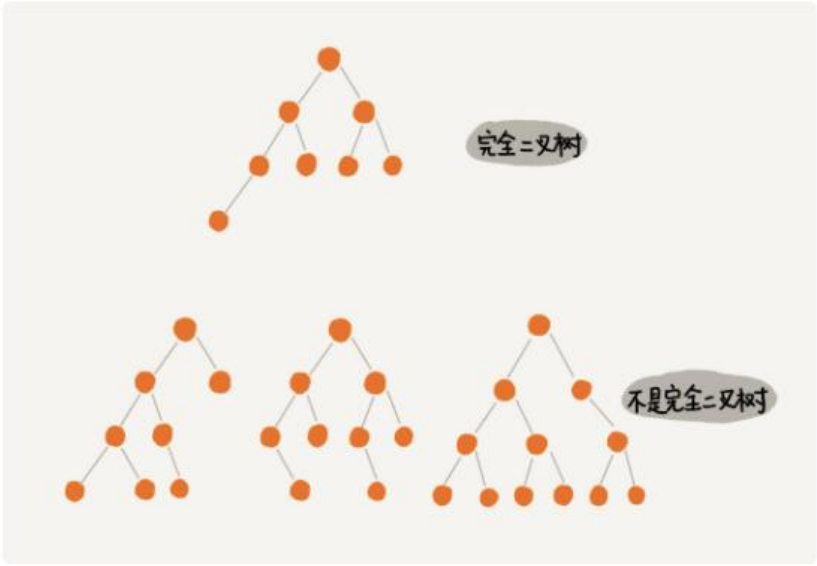
除叶子节点外，每个节点最多只有两个子节点，分别是左子节点和右子节点，可以只有其中一个子节。

满二叉树

叶子节点全都在最底层，除了叶子节点之外，每个节点都有左右两个子节点

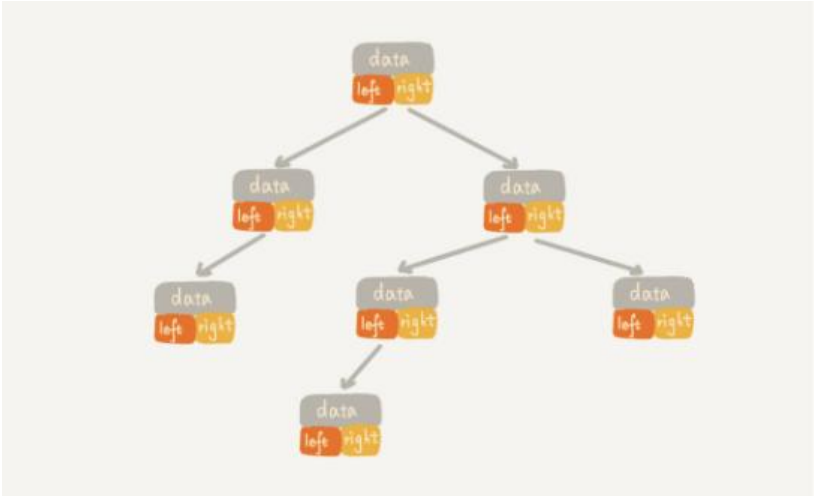
完全二叉树

叶子节点都在最底下两层，**最后一层的叶子节点都靠左排列**，并且除了最后一层，其他层的节点个数要达到最大，不像满二叉树必须要有叶子节点，但是如果有必须是靠左排列。



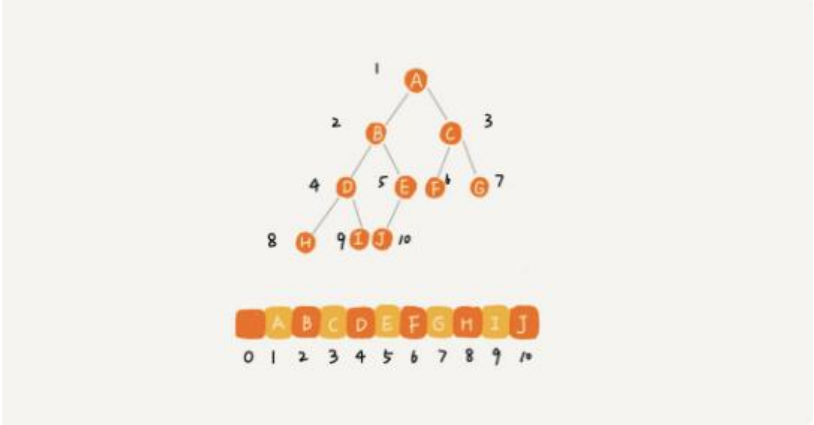
二叉树的存储方式

基于指针或者引用的二叉链式存储法



每个节点有三个字段，一个存储数据，另外两个指向左右子节点的指针，只要有根节点，就可以通过右子节点的指针，把整棵树串起来，这也是比较常用的一种方式。

基于数组的顺序存储法



原文链接：用数组存储二叉树合适么

根节点存储在下标 i = 1 的位置，那左子节点存储在下标 2 * i = 2 的位置，右子节点存储在 2 * i + 1 3 的位置。以此类推，B 节点的左子节点存储在 2 * i = 2 * 2 = 4 的位置，右子节点存储在 2 * i + 1 2 * 2 + 1 = 5 的位置。**情况下，为了方便计算子节点，根节点会存储在下标为 1 的位置**,反过来也能过子节点获取父节点的位置i/2(取整).

我们知道数组申请的是一块连续的内存空间，但是对于普通的二叉树，他不一定是连续存储的，对于间一些节点，可能只有一个子节点，这样就会浪费很多的内存，但是对于完全二叉树，则不会浪费，为除叶子节点外都是两个子节点，叶子节点也是靠左的，能保证连续性，所以用数组存储完全二叉树非常合适的。

二叉树的遍历

二叉树遍历的方法有三种，前序遍历、中序遍历和后序遍历。其中，前、中、后序，表示的是节点与的左右子树节点遍历打印的先后顺序

- 前序遍历是指，对于树中的任意节点来说，先打印这个节点，然后再打印它的左子树，最后打印它右子树。
- 中序遍历是指，对于树中的任意节点来说，先打印它的左子树，然后再打印它本身，最后打印它的子树。
- 后序遍历是指，对于树中的任意节点来说，先打印它的左子树，然后再打印它的右子树，最后打印个节点本身

三种遍历方式的代码实现:

```
public void preTraversal(TreeNode node){
        if (node != null){
            visitNode(node);
            preTraversal(node.leftChild);
            preTraversal(node.rightChild);
        }
    }

    public void traversal(TreeNode node){
        if (node != null){
            traversal(node.leftChild);
            visitNode(node);
            traversal(node.rightChild);
        }
    }

    public void postTraversal(TreeNode node){
        if (node != null){
            traversal(node.leftChild);
            traversal(node.rightChild);
            visitNode(node);
        }
    }
```

每个节点最多会被访问两次，所以遍历操作的时间复杂度，跟节点的个数 n 成正比，也就是说二叉树历的时间复杂度是 O(n).

总结一下

几个比较常用的概念，根节点、叶子节点、父节点、子节点、兄弟节点，还有节点的高度、深度、层，以及树的高度。

平时最常用的树就是二叉树。二叉树的每个节点最多有两个子节点，分别是左子节点和右子节点。二树中，有两种比较特殊的树，分别是满二叉树和完全二叉树。满二叉树又是完全二叉树的一种特殊情。

二叉树既可以用链式存储，也可以用数组顺序存储。数组顺序存储的方式比较适合完全二叉树，其他型的二叉树用数组存储会比较浪费存储空间。除此之外，二叉树里非常重要的操作就是前、中、后序历操作，遍历的时间复杂度是 O(n)。