



链滴

算法基础 (六)--- 树和二叉树 概念梳理

作者: [down-to-earth1994](#)

原文链接: <https://ld246.com/article/1576417586152>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

- 什么是树?

树(tree)是 n ($n \geq 0$)个节点的有限集合。当 $n = 0$ 时, 称为空树。在任意一个非空树中有如下特点。

- 1.有且仅有一个特点的称为根节点。

- 2.当 $n > 1$ 时,其余节点 可分为 m ($m > 0$) 个互不相交的有限集, 每一个集合本身又是一个树, 并为根的子树。

- 什么是叶子节点(leaf).

- 没有孩子 被称为叶子节点 (leaf)

- 什么谁树的高度;

- 树的最大层级数, 被称为树的高度或深度;

- 父节点, 孩子节点, 兄弟节点, 请自行百度;

- 什么是二叉树?

- 这种树的每个节点最多 只有两个孩子节点; 一个左孩子 (left child) , 一个是右孩子(right child)

- 什么是满二叉树?

- 一个二叉树的所以非叶子节点都存在左右孩子, 并且所有叶子节点都在同一个层级上, 那么这棵树就是满二叉树;

- 二叉树存储结构?

- 链式存储;

- 存储数据的data 变量
- 指向左孩子的left 指针

- 指向右孩子的right 指针

- 数组

- 使用数组存储的时候, 按照层级属性把二叉树的节点放到数组中对应的我位置上, 如果某个点的左孩子或者右孩子空缺, 则相应的位置也要空出来; (这样设计是为了更方便的在数组中定位二的孩子节点和父节点)

那么左孩子节点的下标 = $2 * \text{parent} + 1$;右孩子的节点下标 = $2 * \text{parent} + 2$

说明: (对于一个稀疏的二叉树, 用数组维护是相当浪费空间的)