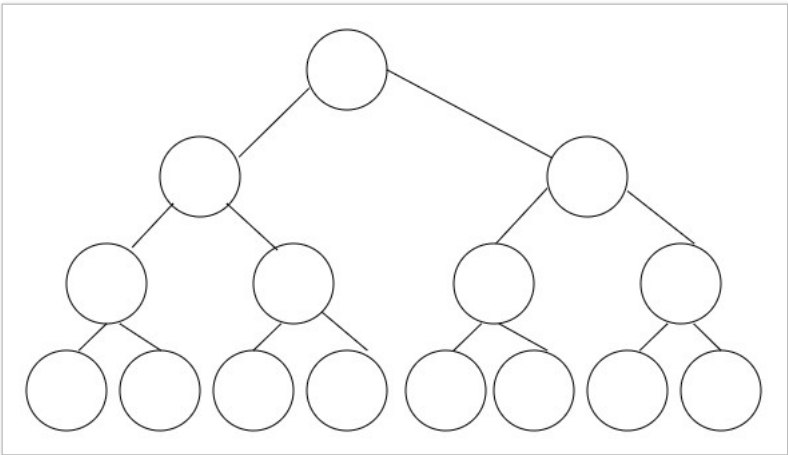


# 1.2 二叉树的基本概念 (下)\_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 1.2 二叉树的基本概念 (下) 涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

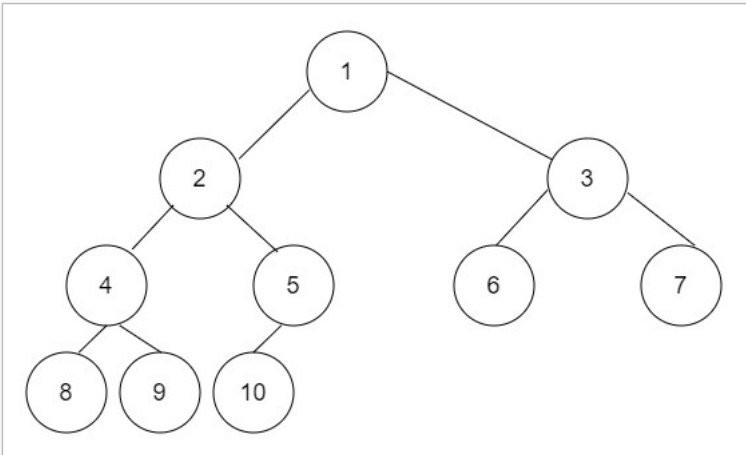
在一颗二叉树中，除了最后一层之外，如果所有的分支节点都存在左子树和右子树，并且所有的叶子节点都在同一层上 \*\*，这样的二叉树，我们称之为满二叉树。 \*\*



满二叉树的特点：

- 1. 叶子节点只会出现在最下面一层。
- 2. 非叶子节点的节点，拥有子树的个数一定为 2.
- 3. 在同样深度的二叉树中，满二叉树的节点个数最多。

对一颗具有 n 个结点的二叉树按层进行编号，如果编号为 i (1 <= i <= n) 的结点与同样深度的满二叉树节点编号为 i 的结点在二叉树中的位置完全相同，则这颗树，我们称之为完全二叉树。



完全二叉树的特点

思考：当一个结点的编号为 i,n 表示二叉树总的结点个数：

i = 2, 左孩子为 4 , 右孩子为 5

i = 3, 左孩子为 6, 右孩子为 7

l = 6, 左右孩子都为空

总结:

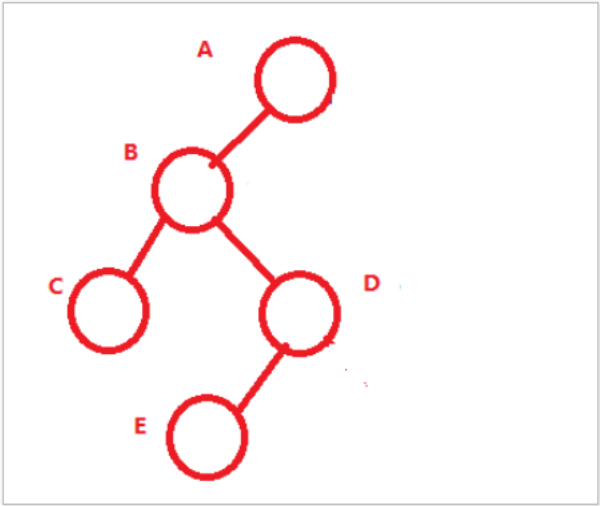
对于完全二叉树, 共有 n 个结点, 编号为 i(i>=1) 的结点:

(1) 左孩子存在条件:  $2 * i \leq n$ (节点的个数), 左孩子编号为  $2 * i$

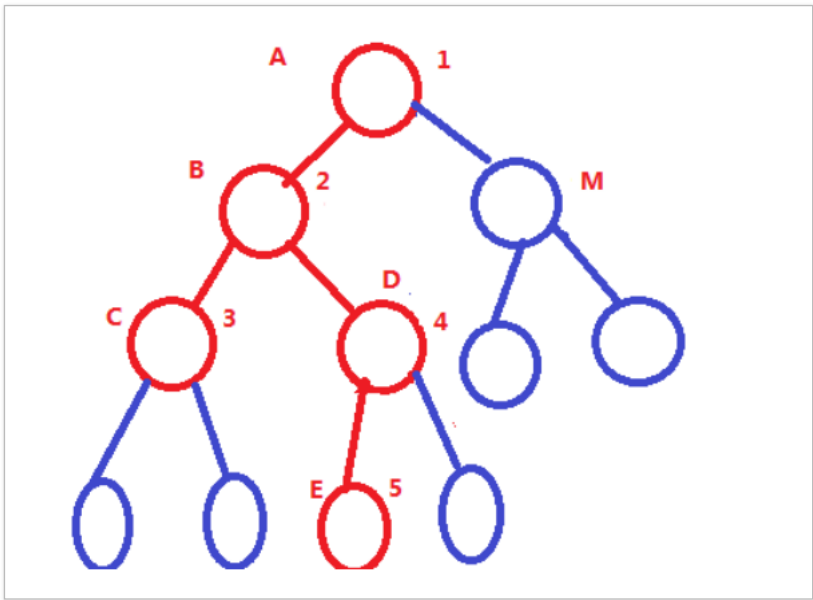
(2) 右孩子存在条件:  $2 * i + 1 \leq n$ , 右孩子编号  $2 * i + 1$

(顺序存储的话, 若不是完全二叉树存储没有意义。)

假设下面有一颗树, 我们如何把它存到数组中呢?



思路: 先把转换成完全二叉树, 然后再编号。

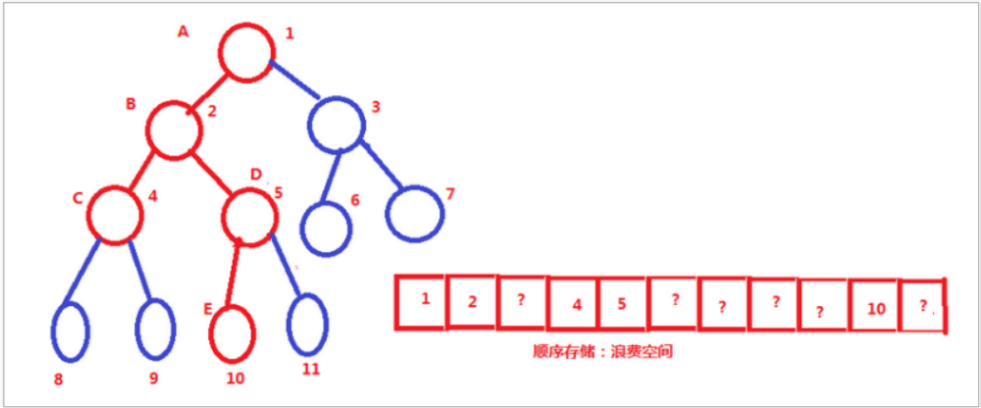


这样存储就看似没有什么问题 \*\*。我们可以按照编号把数据存储到数组中, 我们按照编号 (1,2,3,4,5) 的顺序存储就可以了啊! 这个时候, 我就要问了, 假说说, 我们的 m 的编号, 你怎么知道我们的 3 好位置是在下面, 而不是在我们的 m 编号的位置呢? 我们的连续存储无法识别。(这种方法, 我们无法推断树的结构)。\*\*

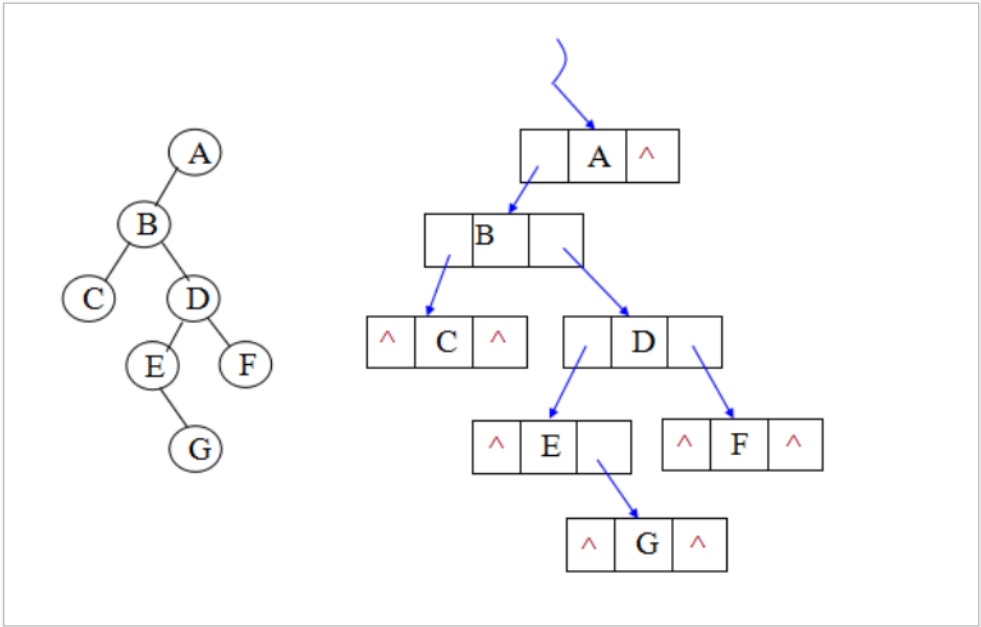
因此, 我们顺序存储规定:

无论是何种树, 我们都会转换成完全二叉树。然后一层一层

的从左给我们的二叉树进行编号，然后存储在数组中。及如下图。



链式存储：定义结点保存左孩子和右孩子的地址。



```
typedef char data_t;

typedef struct bnode
{
    data_t data;
    struct bnode * lchild;
    struct bnode * rchild;
}bitree_t;
```

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

