

数据结构和算法

作者: 小甲鱼

让编程改变世界

Change the world by program

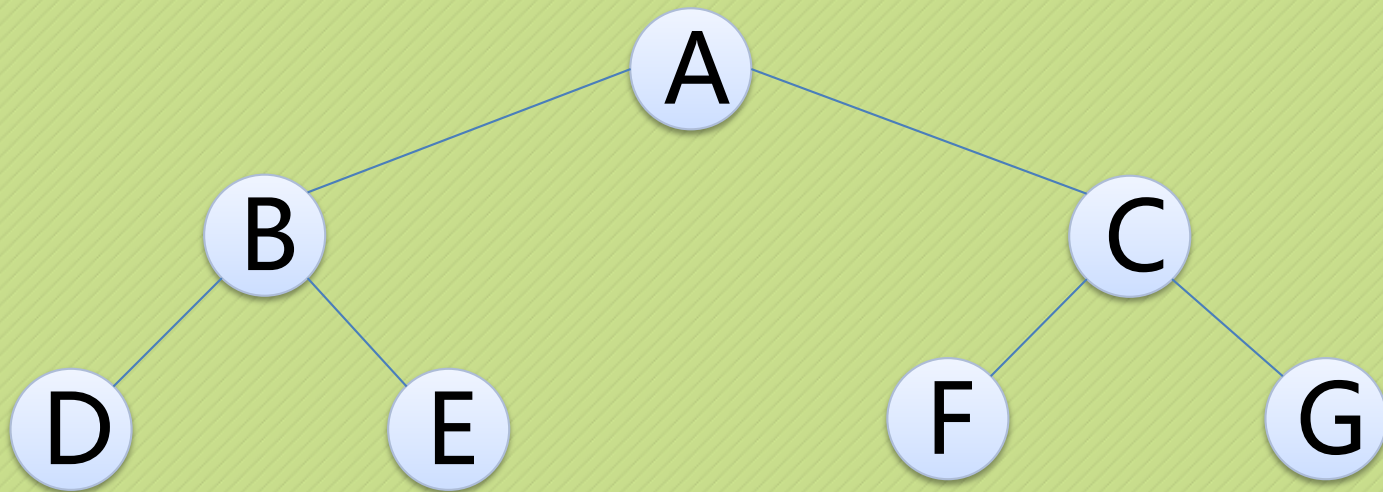


二叉树的存储结构

- 树结构在计算机中的存储形式很多, 可谓天马行空任你创造, 只要能够按照要求完成任务即可。
- 在前边的演示中, 我们发觉很难单单只用顺序存储结构或者链式存储结构来存放。
- 但是二叉树是一种特殊的树, 由于它的特殊性, 使得用顺序存储结构或链式存储结构都能够简单实现。
- 二叉树的顺序存储结构就是用一维数组存储二叉树中的各个结点, 并且结点的存储位置能体现结点之间的逻辑关系。
- 这一点儿都不难: No pic you say a J8!

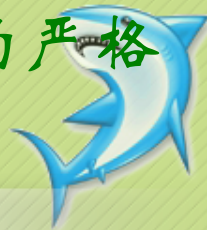


二叉树的顺序存储结构



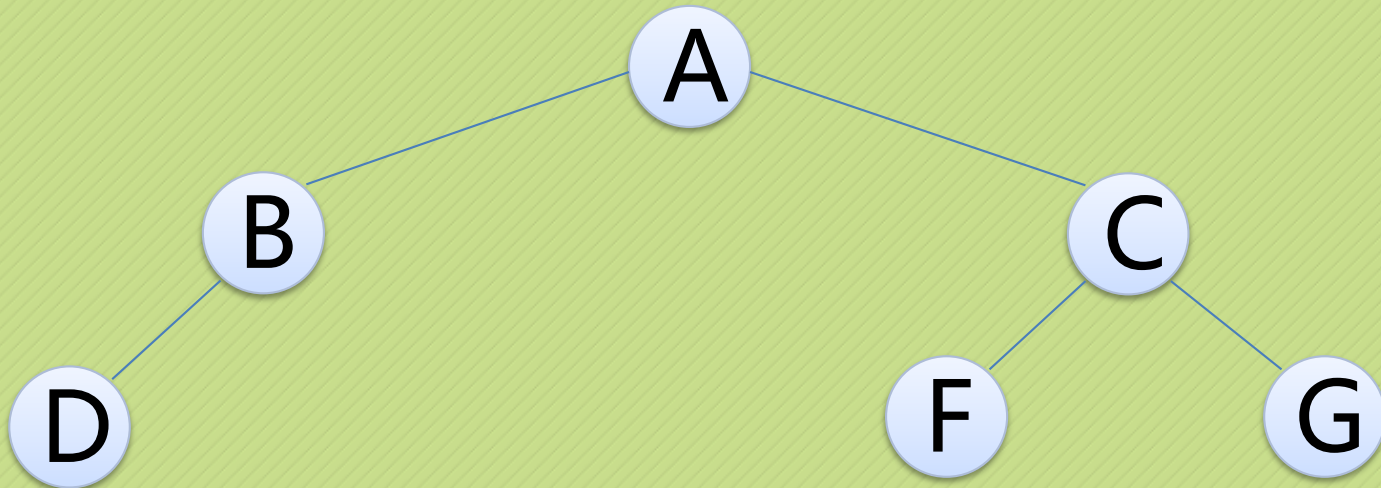
元素	A	B	C	D	E	F	G
下标	1	2	3	4	5	6	7

- 这下看出完全二叉树的优越性来了吧？由于他的严格定义，在数组直接能表现出逻辑结构。



二叉树的顺序存储结构

- 当然对于一般的二叉树，尽管层序编号不能反映逻辑关系，但是也可以按照完全二叉树编号方式修改一下，把不存在的结点用“^”代替即可。

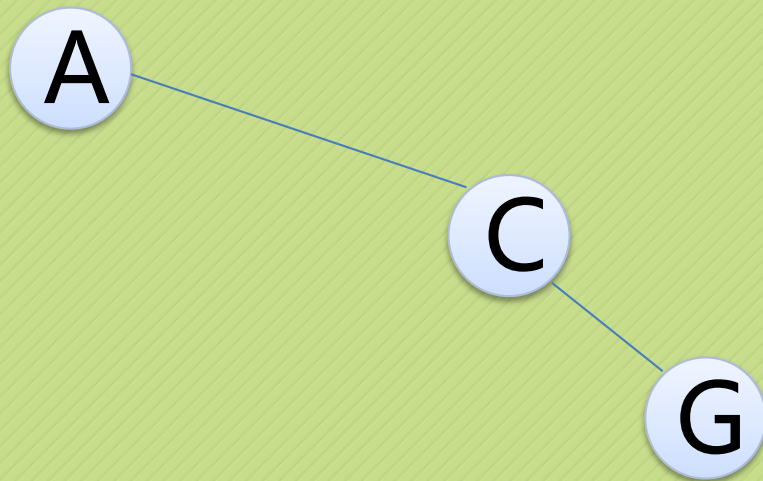


元素	A	B	C	D	^	F	G
下标	1	2	3	4	5	6	7



二叉树的顺序存储结构

- 但是考虑到一种极端的情况，回顾一下斜树，如果是一个又斜树，那么会变成这样。。。。。



元素	A	^	C	^	^	^	G
下标	1	2	3	4	5	6	7



二叉链表

- 既然顺序存储方式的适用性不强，那么我们就要考虑链式存储结构啦。二叉树的存储按照国际惯例来说一般也是采用链式存储结构的。
- 二叉树每个结点最多有两个孩子，所以为它设计一个数据域和两个指针域是比较自然的想法，我们称这样的链表叫做二叉链表。

lchild	data	rchild
--------	------	--------

- 以下是我们的二叉链表的结点结构定义代码：



二叉链表

```
typedef struct BiTNode
{
    ElemType data;
    struct BiTNode *lchild, *rchild;
} BiTNode, *BiTree;
```

