

二叉树-5

- 案例：N 个节点不同二叉树个数

- 概述：

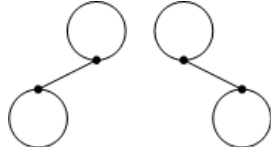
- 思路：

- 节点个数

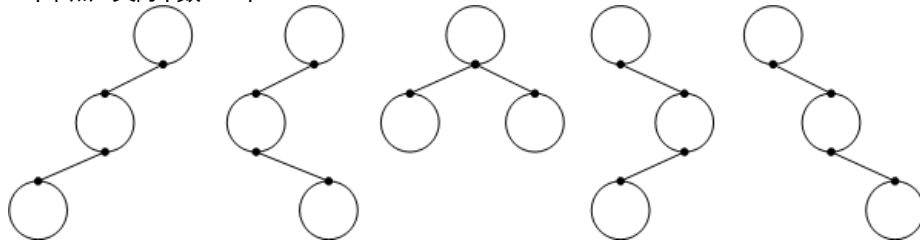
- 0 个节点二叉树个数：注意，这里是 1 个，代表空树

- 1 个节点二叉树个数：1 个

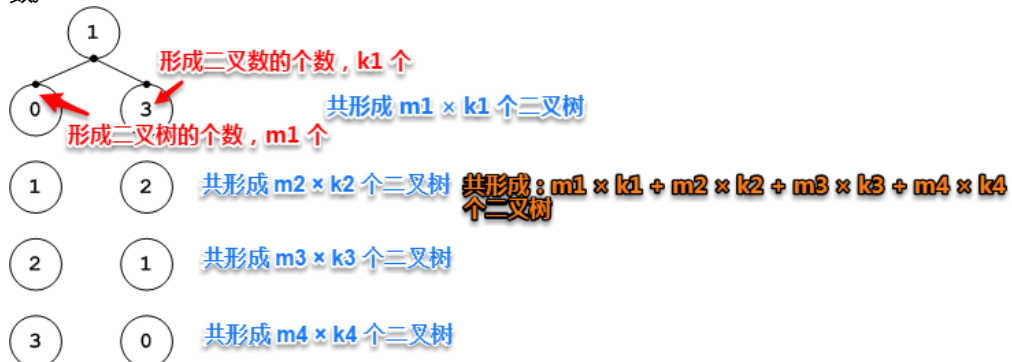
- 2 个节点二叉树个数：2 个



- 3 个节点二叉树个数：5 个



- 计数思路：递归问题。一次分配形成二叉树的个数等于分配给左子树的节点能够形成的左子树个数 \times 分配给右子树的节点能够形成的右子树个数， n 个节点共有 $n - 1$ 种分配情况，每种分配情况求合得到 n 个节点能够形成的二叉树个数。



- 代码

- 代码-1：

```
# coding:utf-8

def count (n):
    # 左子树节点个数 k ∈ (0, n-1)
    # 右子树节点个数 n - k - 1 ∈ (0, n-1)

    if n == 0: # 0 个节点形成空树 相当于 1 个二叉树
        return 1
    sum = 0
    for k in xrange(n):
        sum = sum + count (k) * count (n - 1 - k) # 一次分配
                                                # 左子树 k 个节点
                                                # 右子树 n - 1 - k 个节点

    return sum

if __name__ == '__main__':
    print count (4)
```

- 代码-2

- And So On

