**以2个a、1个b为例：**

向左走，表示增加一个a；

向右走，表示增加一个b.

注意看两组橙色和绿色节点组（相同颜色为一组），我们发现，左边的节点就是把右边的节点平移过来的，就是把右边的节点叠加到a节点上去。这是我们动态规划填表的基石。

动态规划：dp[i,j]表示到目前为止，由i个’a’和j个’b’组成的字典树的节点总数（包括根节点，节点没有字母，表示为’0’，如上图）。

**初始化：**

第0列：表示分别有0、1、2、3、4个’a’字符组成的字典树，那么就是二叉树一直向左走，节点数由1一直增加到5.

第0行：与上面一样，二叉树一直向右走。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | b | b | b | b | b |
|  |  | 0 | 1 | 2 | 3 | 4 | 5 |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| a | 1 | 2 | 1+2+2=5 |  |  |  |  |
| a | 2 | 3 |  |  |  |  |  |
| a | 3 | 4 |  |  |  |  |  |
| a | 4 | 5 |  |  |  |  |  |

第1行：

dp[i,j]=1+dp[i-1,j]+dp[i,j-1]，原因如下图。

图1

图2

图3

1. 起始只有一个根节点，

dp[i,j]初始化为1：

dp[i,j]=1;

**（2）**在a节点上加上右边的

子树（0和b），得到灰色的

三个节点：

dp[i,j]+=dp[i-1,j];

**（3）**在b节点上，加上图1子树

（0和a），于是新增2个黄色节点：

dp[i,j]+=dp[i,j-1].