

通过先序和中序数组生成后序数组

【题目】

已知一棵二叉树所有的节点值都不同，给定这棵树正确的先序和中序数组，不要重建整棵树，而是通过这两个数组直接生成正确的后序数组。

【难度】

士 ★☆☆☆

【解答】

举例说明生成后序数组的过程，假设 $pre=[1,2,4,5,3,6,7]$ ， $in=[4,2,5,1,6,3,7]$ 。

1. 根据 pre 和 in 的长度，生成长度为 7 的后序数组 pos ，按以下规则从右到左填满 pos 。
2. 根据 $[1,2,4,5,3,6,7]$ 和 $[4,2,5,1,6,3,7]$ ，设置 $pos[6]=1$ ，即先序数组最左边的值。根据 1 把 in 划分成 $[4,2,5]$ 和 $[6,3,7]$ ， pre 中 1 的右边部分根据这两部分等长划分出 $[2,4,5]$ 和 $[3,6,7]$ 。 $[2,4,5]$ 和 $[4,2,5]$ 一组， $[3,6,7]$ 和 $[6,3,7]$ 一组。
3. 根据 $[3,6,7]$ 和 $[6,3,7]$ ，设置 $pos[5]=3$ ，再次划分出 $[6]$ （来自 $[3,6,7]$ ）和 $[6]$ （来自 $[6,3,7]$ ）一组， $[7]$ （来自 $[3,6,7]$ ）和 $[7]$ （来自 $[6,3,7]$ ）一组。
4. 根据 $[7]$ 和 $[7]$ 设置 $pos[4]=7$ 。
5. 根据 $[6]$ 和 $[6]$ 设置 $pos[3]=6$ 。
6. 根据 $[2,4,5]$ 和 $[4,2,5]$ ，设置 $pos[2]=2$ ，再次划分出 $[4]$ （来自 $[2,4,5]$ ）和 $[4]$ （来自 $[4,2,5]$ ）一组， $[5]$ （来自 $[2,4,5]$ ）和 $[5]$ （来自 $[4,2,5]$ ）一组。
7. 根据 $[5]$ 和 $[5]$ 设置 $pos[1]=5$ 。
8. 根据 $[4]$ 和 $[4]$ 设置 $pos[0]=4$ 。

如上过程简单总结为：根据当前的先序和中序数组，设置后序数组最右边的值，然后划分出左子树的先序、中序数组，以及右子树的先序、中序数组，先根据右子树的划分设置好后序数组，再根据左子树的划分，从右边到左边依次设置好后序数组的全部位置。

具体过程请参看如下代码中的 `getPosArray` 方法。

```
public int[] getPosArray(int[] pre, int[] in) {  
    if (pre == null || in == null) {  
        return null;  
    }  
}
```

```

    }
    int len = pre.length;
    int[] pos = new int[len];
    HashMap<Integer, Integer> map = new HashMap<Integer, Integer>();
    for (int i = 0; i < len; i++) {
        map.put(in[i], i);
    }
    setPos(pre, 0, len - 1, in, 0, len - 1, pos, len - 1, map);
    return pos;
}

// 从右往左依次填好后序数组 s
// si 为后序数组 s 该填的位置
// 返回值为 s 该填的下一个位置
public int setPos(int[] p, int pi, int pj, int[] n, int ni, int nj,
    int[] s, int si, HashMap<Integer, Integer> map) {
    if (pi > pj) {
        return si;
    }
    s[si--] = p[pi];
    int i = map.get(p[pi]);
    si = setPos(p, pj - nj + i + 1, pj, n, i + 1, nj, s, si, map);
    return setPos(p, pi + 1, pi + i - ni, n, ni, i - 1, s, si, map);
}

```

统计和生成所有不同的二叉树

【题目】

给定一个整数 N ，如果 $N < 1$ ，代表空树结构，否则代表中序遍历的结果为 $\{1, 2, 3, \dots, N\}$ 。请返回可能的二叉树结构有多少。

例如， $N = -1$ 时，代表空树结构，返回 1； $N = 2$ 时，满足中序遍历为 $\{1, 2\}$ 的二叉树结构只有如图 3-49 所示的两种，所以返回结果为 2。

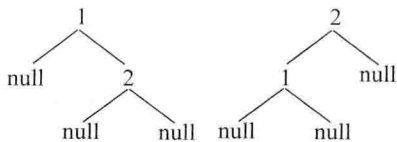


图 3-49

进阶： N 的含义不变，假设可能的二叉树结构有 M 种，请返回 M 个二叉树的头节点，每一棵二叉树代表一种可能的结构。