1. 题目
2. 题干

给定一个非负整数 numRows，生成杨辉三角的前 numRows 行。

在杨辉三角中，每个数是它左上方和右上方的数的和。

1. 示例

**输入:** 5

**输出:**

[

[1],

[1,1],

[1,2,1],

[1,3,3,1],

[1,4,6,4,1]

]

1. 题解
2. 思路

杨辉三角的题目给我们思路，由上一行的相邻数相加得到下一行的数。进而仔细观察，除了第一行之外，其他的每一行的前后数字都是1，而通过前一行相加得到的数的数量为上一行的数字数量减一。因此，我们可以通过迭代获取上一行的链表，然后两两相加相邻的数得到下一行相应的数。

注意，每一行的链表都是需要新建的，然后再添加到总的链表中。

1. 代码实现

Java：

class Solution {

    public List<List<Integer>> generate(int numRows) {

        List<List<Integer>> out = new ArrayList<List<Integer>>();

        if(numRows == 0){

            return out;

        }else{

            out.add(new ArrayList<Integer>());

            out.get(0).add(1);

            for(int i = 1; i < numRows; ++i){

                List<Integer> row = new ArrayList<Integer>();

                row.add(1);

                List<Integer> rowPre = out.get(i-1);

                for(int j = 1; j < i; ++j){

                    row.add(rowPre.get(j-1) + rowPre.get(j));

                }

                row.add(1);

                out.add(row);

            }

            return out;

        }

    }

}