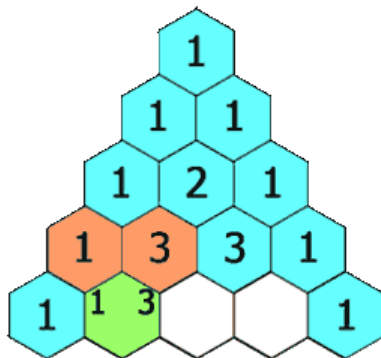


119. Pascal's Triangle II

[Description](#)[Hints](#)[Submissions](#)[Discuss](#)[Solution](#)[Pick One](#)

Given a non-negative index k where $k \leq 33$, return the k^{th} index row of the Pascal's triangle.

Note that the row index starts from 0.



In Pascal's triangle, each number is the sum of the two numbers directly above it.

Example:

Input: 3
Output: [1,3,3,1]

Follow up:

Could you optimize your algorithm to use only $O(k)$ extra space?

```

public class L119 {
    public List<Integer> getRow(int rowIndex) {
        List<Integer> list = new ArrayList<Integer>();

        if(rowIndex < 0)
            return list;

        for(int i = 0; i <= rowIndex; i++) {
            list.add(1);
        }

        if(rowIndex == 0 || rowIndex == 1)
            return list;

        int i = 2;
        while (i <= rowIndex) {
            //这里必须从后面开始，因为从前面开始会修改原来的数，导致不是加的原来的数
            for(int j = i - 1; j >= 1; j--) {
                list.set(j, list.get(j-1) + list.get(j));
            }
            i++;
        }
        return list;
    }

    public static void main(String [] args) {
        new L119().getRow(4);
    }
}

```