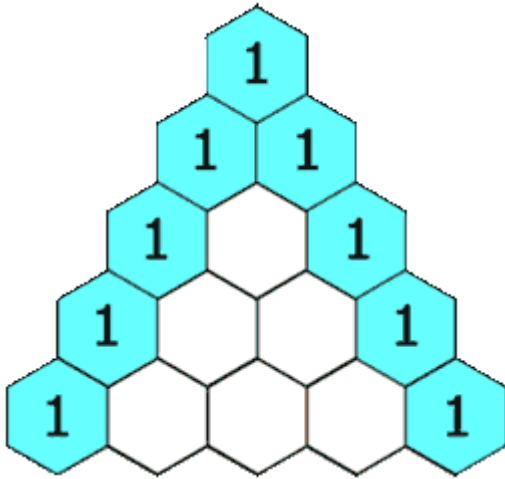


119. Pascal's Triangle II

Given an integer `rowIndex`, return the `rowIndexth` (**0-indexed**) row of the **Pascal's triangle**.

In **Pascal's triangle**, each number is the sum of the two numbers directly above it as shown:



Example 1:

Input: `rowIndex = 3`

Output: `[1,3,3,1]`

Example 2:

Input: `rowIndex = 0`

Output: `[1]`

Example 3:

Input: `rowIndex = 1`

Output: `[1,1]`

Constraints:

- `0 <= rowIndex <= 33`

Follow up: Could you optimize your algorithm to use only `O(rowIndex)` extra space?

```
class Solution:
    def getRow(self, rowIndex: int) -> List[int]:
        val = 1
        ans = []
        for i in range(rowIndex+1):
```

```
    ans.append(val)
    val = val*(rowIndex-i)/(i+1)
return ans
```