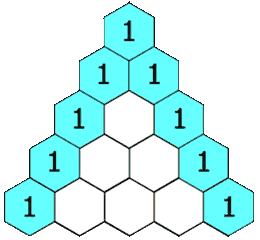
**119. Pascal's Triangle II: -**

Easy Accepted: 769.5K Submissions: 1.2M Acceptance Rate: 62.5%

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?

**Code: -**

class Solution {

public:

    vector<int> getRow(int row) {

        int prev, temp;

        vector<int> ans(row + 1);

        ans[0] = 1;

        for(int i=1; i<=row; ++i){

            prev = 0;

            for(int j=0; j<=i; ++j){

                temp = ans[j];

                ans[j] = ans[j] + prev;

                prev = temp;

            }

        }

        return ans;

    }

};

**T.C: - O(N2)**

**S.C: - O(1) excluding answer storage**