

# Problem 119: Pascal's Triangle II

## Problem Information

**Difficulty:** Easy

**Acceptance Rate:** 66.73%

**Paid Only:** No

**Tags:** Array, Dynamic Programming

## Problem Description

Given an integer `rowIndex`, return the `rowIndex`<sup>th</sup> (\*\*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 Snippets

### C++:

```
class Solution {  
public:  
    vector<int> getRow(int rowIndex) {  
  
    }  
};
```

### Java:

```
class Solution {  
public List<Integer> getRow(int rowIndex) {  
  
}  
}
```

### Python3:

```
class Solution:  
    def getRow(self, rowIndex: int) -> List[int]:
```