119 Pascal's Triangle II

2018年4月1日 15:32

Question:

Given an index *k*, return the k^{th} row of the Pascal's triangle.
For example, given *k* = 3,
Return [1,3,3,1].
Note:
Could you optimize your algorithm to use only *O*(*k*) extra space?

来自 <https://leetcode.com/problems/pascals-triangle-ii/description/>

给定一个索引 *k*，返回帕斯卡三角形（杨辉三角）的第 k 行。

例如，给定 k = 3，

则返回 [1, 3, 3, 1]。

注：

你可以优化你的算法到 *O*(*k*) 的空间复杂度吗?

Solution for Python3:

```python
class Solution:
    def getRow(self, rowIndex):
        """
        :type rowIndex: int
        :rtype: List[int]
        """
        triangle = [0 for _ in range(rowIndex + 1)]
        triangle[0] = 1
        for i in range(1, rowIndex + 1):
            for j in range(i, 0, -1):
                triangle[j] += triangle[j - 1]
        return triangle
```

Solution for C++:

```cpp
class Solution {
```

```
public:
    vector<int> getRow(int rowIndex) {
        vector<int> triangle(rowIndex + 1, 0);
        triangle[0] = 1;
        for (int i = 1; i <= rowIndex; i++) {
            for (int j = i; j >= 1; j--) {
                triangle[j] += triangle[j - 1];
            }
        }
        return triangle;
    }
};
```

Appendix:

每一行的尾部元素是前一行当前列元素+前一行前一列元素。