Pascal's Triangle II

Question: Given an index k, return the kth 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?

Solutions:

```
class Solution:
    # @return a list of integers
    def getRow(self, rowIndex):
        if rowIndex == 0:       return [1]
        if rowIndex == 1:       return [1, 1]
        result = [1]
        nextDivisor = 1
        nextMultiplier = rowIndex
        for _ in range(rowIndex // 2):
            nextVal = int((result[-1] * nextMultiplier) / nextDivisor)
            result.append(nextVal)
            nextDivisor += 1
            nextMultiplier -= 1
        if rowIndex % 2 == 1:   result.extend(result[::-1])
        else:                   result.extend(result[-2::-1])

        return result

Solution().getRow(3)
```