Leetcode Problem 1. (Easy)

Pascal's Triangle

Given an integer numRows, return the first numRows of **Pascal's triangle**.

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



**Example 1:**

**Input:** numRows = 5

**Output:** [[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]]

**Example 2:**

**Input:** numRows = 1

**Output:** [[1]]

**Constraints:**

* 1 <= numRows <= 30

Link: <https://leetcode.com/problems/pascals-triangle/>

class Solution {

public List<List<Integer>> generate(int numRows) {

List<List<Integer>> triangle = new ArrayList<>();

if (numRows == 0) {

return triangle;

}

triangle.add(new ArrayList<>());

triangle.get(0).add(1);

for (int i = 1; i < numRows; i++) {

List<Integer> row = new ArrayList<>();

List<Integer> prevRow = triangle.get(i - 1);

row.add(1);

for (int j = 1; j < i; j++) {

int sum = prevRow.get(j - 1) + prevRow.get(j);

row.add(sum);

}

row.add(1);

triangle.add(row);

}

return triangle;

}

}

