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

So I am assuming that we all are aware of what is Pascal’s Triangle, if not then visit [Pascal's Triangle](https://www.geeksforgeeks.org/pascal-triangle/).

So now that we know Pascal’s Triangle, lets understand how many elements will be there in each row.

For 0th row there is 1 element.

For 1st row there are 2 elements.

For 2nd row there are 3 elements.

Therefor, for nth row there will be n+1 elements

Now the base case is for 0th row there is only 1 element in the sub-array, i.e., 1.

For all other rows the 0th column and last column’s value will be 1. And we will be left with n-2 elements to be calculated.

Those elements would be the sum of previous row’s ith column value and (i-1)th column value.

Code:  
class Solution {

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

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

int i=1;

if(numRows==0){

return dp;

}

while(i<=numRows){

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

if(i==1){

subprblm.add(1);

}else{

subprblm.add(1);

for(int k=1; k<i-1;k++){

subprblm.add(dp.get(i-2).get(k-1)+dp.get(i-2).get(k));

}

subprblm.add(1);

}

dp.add(subprblm);

i++;

}

return dp;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>