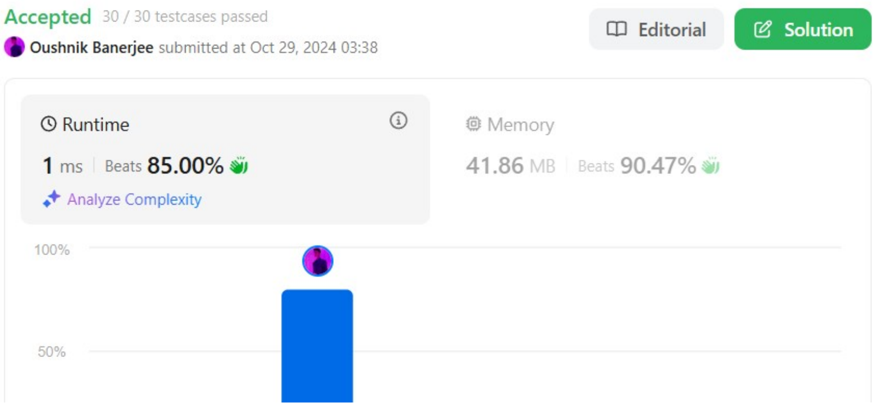
AP Experiment 10

Pascal's Triangle

```
class Solution {
    public List<List<Integer>> generate(int numRows) {
        List<List<Integer>> triangle= new ArrayList<List<Integer>>();
        if(numRows== 0){
            return triangle;
        }
        ArrayList<Integer> prev= new ArrayList<Integer>();
        prev.add(1);
        triangle.add(prev);
        for(int i=2; i<= numRows; i++){
            ArrayList<Integer> present= new ArrayList<Integer>();
            present.add(1);
            for(int j=0; j< prev.size()-1; j++){
                present.add(prev.get(j)+ prev.get(j+1));
            }
            present.add(1);
            triangle.add(present);
            prev= present;
        }

        return triangle;
    }
}
```



Accepted 30 / 30 testcases passed

Oushnik Banerjee submitted at Oct 29, 2024 03:38

Runtime

1 ms | Beats 85.00%

Memory

41.86 MB | Beats 90.47%

Hamming Distance

```
class Solution {
    public int hammingDistance(int x, int y) {
        return Integer.bitCount(x ^ y);
    }
}
```