Discover. Learn. Empower. ENGINEERING

WORKSHEET-10

Student Name: Meenansh Gupta **UID:** 22BCS16380

Branch: CSE **Section/Group:** NTPP-603-B

Semester: 6th **Date of Performance:** 15/04/25

Subject Name: AP-2 **Subject Code:** 22CSP-351

Aim(i): *Given an integer numRows, return the first numRows of Pascal's triangle.*

Source Code:

```
class Solution {
public:
    vector<vector<int>> generate(int numRows) {
        vector<vector<int>> result;
        vector<int> prevRow;

        for (int i = 0; i < numRows; i++) {
            vector<int> currentRow(i + 1, 1);

            for (int j = 1; j < i; j++) {
                currentRow[j] = prevRow[j - 1] + prevRow[j];
            }

            result.push_back(currentRow);
            prevRow = currentRow;
        }

        return result;
    }
};
```

OUTPUT:

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

numRows =

5

Output

[[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]]

Expected

[[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]]

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

numRows =

1

Output

[[1]]

Expected

[[1]]

Accepted 30 / 30 testcases passed

Meenansh_16380 submitted at Apr 18, 2025 06:54

Editorial Solution

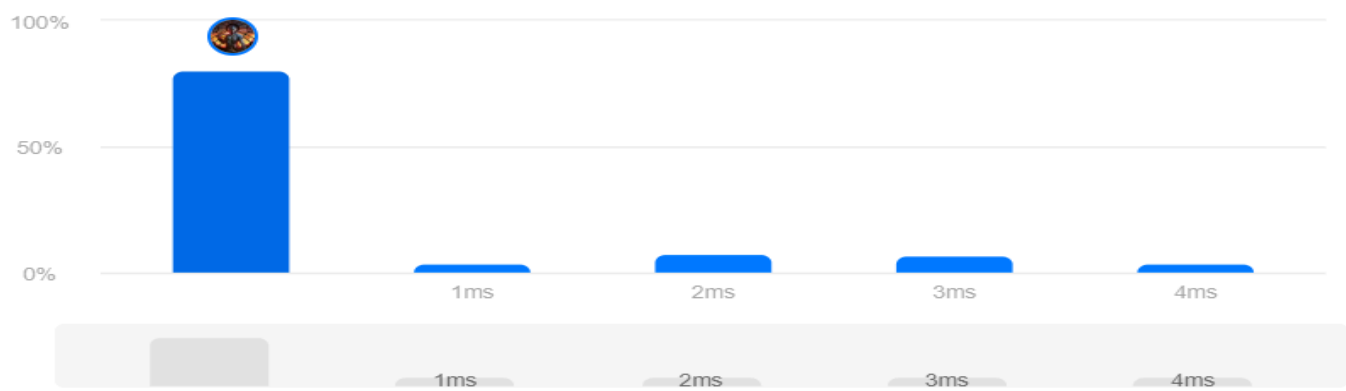
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

9.59 MB | Beats 89.44%



Aim(ii): *Given a positive integer n, write a function that returns the number of set bits in its binary representation (also known as the Hamming weight).*

Source Code:

```
class Solution {
public:
    int hammingWeight(uint32_t n) {
        int res = 0;
        for (int i = 0; i < 32; i++) {
            if ((n >> i) & 1) {
                res += 1;
            }
        }
        return res;
    }
};
```

OUTPUT:

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

n =

11

Output

3

Expected

3

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

n =

128

Output

1

Expected

1

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

n =

2147483645

Output

30

Expected

30

Accepted 598 / 598 testcases passed

Meenansh_16380 submitted at Apr 18, 2025 06:59

Editorial Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

8.35 MB | Beats 11.84%

Aim(iii): *Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it can trap after raining.*

Source Code:

```
class Solution {
public:
    int trap(vector<int>& height) {
        int left = 0;
        int right = height.size() - 1;
        int leftMax = height[left];
        int rightMax = height[right];
        int water = 0;

        while (left < right) {
            if (leftMax < rightMax) {
                left++;
                leftMax = max(leftMax, height[left]);
                water += leftMax - height[left];
            } else {
                right--;
                rightMax = max(rightMax, height[right]);
                water += rightMax - height[right];
            }
        }

        return water;
    }
};
```

OUTPUT:

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

height =

[0,1,0,2,1,0,1,3,2,1,2,1]

Output

6

Expected

6

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

height =

[4,2,0,3,2,5]

Output

9

Expected

9

← All Submissions

Accepted 324 / 324 testcases passed

Meenansh_16380 submitted at Apr 18, 2025 07:00

Editorial Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

25.96 MB | Beats 80.02%

Learning Outcome

1. We learnt about Graphs.
2. We learnt about Dynamic Programming.
3. We learnt about vectors