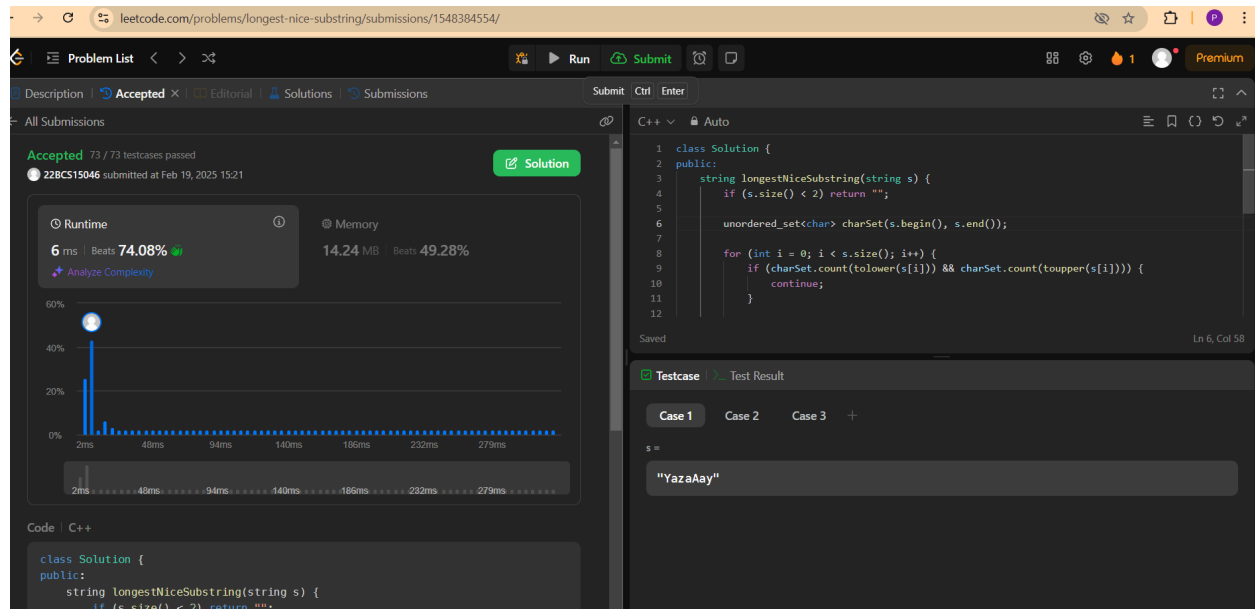


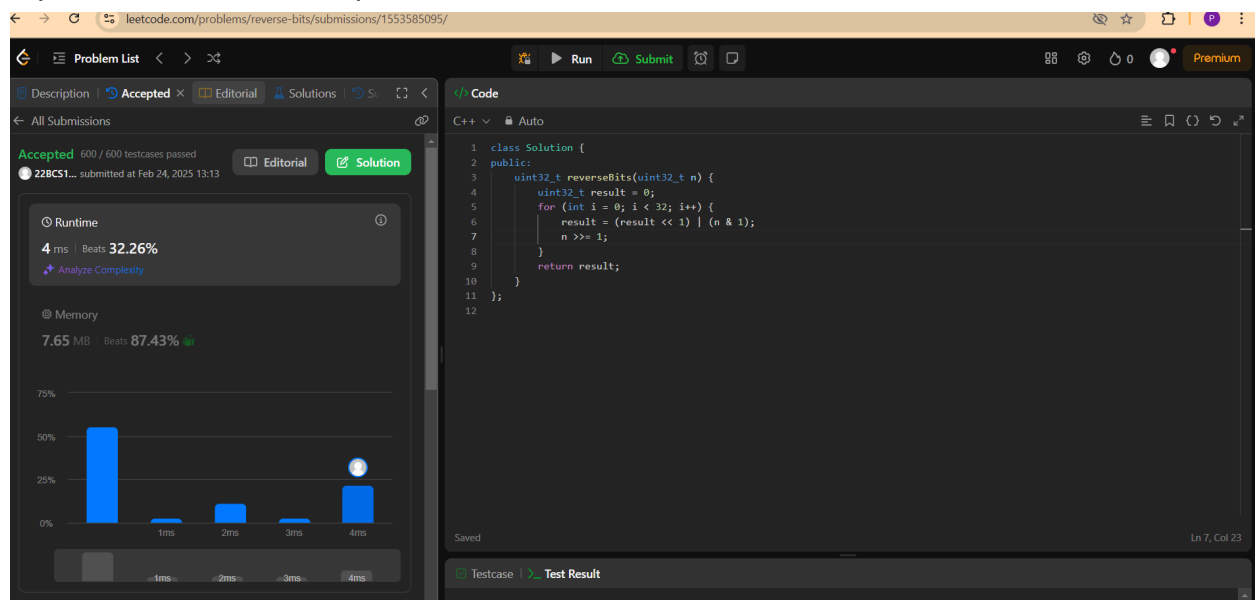
# ASSIGNMENT-4(AP)

## 22BCS15046(PRIYANSHU)

### 1(LONGEST NICE SUBSTRING)



### 2(REVERSE BITS)



### 3(NUMBER OF 1 BITS)

Screenshot of a LeetCode submission for the problem "Number of Bits" (1553589643). The submission is accepted, showing a runtime of 0 ms and memory usage of 8.19 MB. The code is in C++ and implements a function to calculate the number of bits in a given integer.

```
1 class Solution {
2 public:
3     int hammingWeight(int n) {
4         int count = 0;
5         while (n) {
6             count += n & 1;
7             n >>= 1;
8         }
9         return count;
10    };
11 };
12
```

## 4(MAXIMUM SUBARRAY)

Screenshot of a LeetCode submission for the problem "Maximum Subarray" (1553591026). The submission is accepted, showing a runtime of 0 ms and memory usage of 71.77 MB. The code is in C++ and implements a function to find the maximum sum of a contiguous subarray.

```
2 public:
3     int maxSubArray(vector<int>& nums) {
4         int maxSum = nums[0];
5         int currentSum = nums[0];
6
7         for (int i = 1; i < nums.size(); i++) {
8             currentSum = max(nums[i], currentSum + nums[i]);
9             maxSum = max(maxSum, currentSum);
10        }
11
12        return maxSum;
13    }
```

## 5(SEARCH A 2D MATRIX ||)

Screenshot of a LeetCode submission for the problem "Search a 2D Matrix II" (1553597806). The submission is accepted, showing a runtime of 107 ms and memory usage of 18.70 MB. The code is in C++ and implements a function to search for a target in a 2D matrix.

```
1 class Solution {
2 public:
3     bool searchMatrix(vector<vector<int>>& matrix, int target) {
4         int m = matrix.size(), n = matrix[0].size();
5
6         for (int i = 0; i < m; i++) {
7             int left = 0, right = n - 1;
8             while (left <= right) {
9                 int mid = left + (right - left) / 2;
10                if (matrix[i][mid] == target) return true;
11                else if (matrix[i][mid] < target) left = mid + 1;
12                else right = mid - 1;
13            }
14        }
15
16        return false;
17    };
18 };
19
```

# 6(SUPER POW)

leetcode.com/problems/super-pow/

Problem List < > > >

Description | Accepted x | Editorial | Solutions | Submission

< All Submissions

Accepted 57 / 57 testcases passed

22BCS15046 submitted at Feb 24, 2025 13:33

Solution

Runtime

7 ms | Beats 8.16%

Analyze Complexity

Memory

15.35 MB | Beats 15.60%

75%  
50%  
25%  
0%

2ms 4ms 6ms 8ms

Code

```
1 class Solution {
2 public:
3     const int MOD = 1337;
4
5     int powerMod(int x, int y) {
6         int res = 1;
7         x %= MOD;
8         while (y > 0) {
9             if (y % 2 == 1)
10                res = (res * x) % MOD;
11            x = (x * x) % MOD;
12            y /= 2;
13        }
14        return res;
15    }
16
17    int superPow(int a, vector<int>& b) {
18        if (b.empty()) return 1;
19
20        int lastDigit = b.back();
21        b.pop_back();
22
23        int part1 = powerMod(superPow(a, b), 10);
24        int part2 = powerMod(a, lastDigit);
25
26        return (part1 * part2) % MOD;
27    }
28};
```

Saved

Testcase | Test Result

Ln 4, Col 1