ADVANCED PROGRAMMING LAB-2 ASSIGNMENT 4

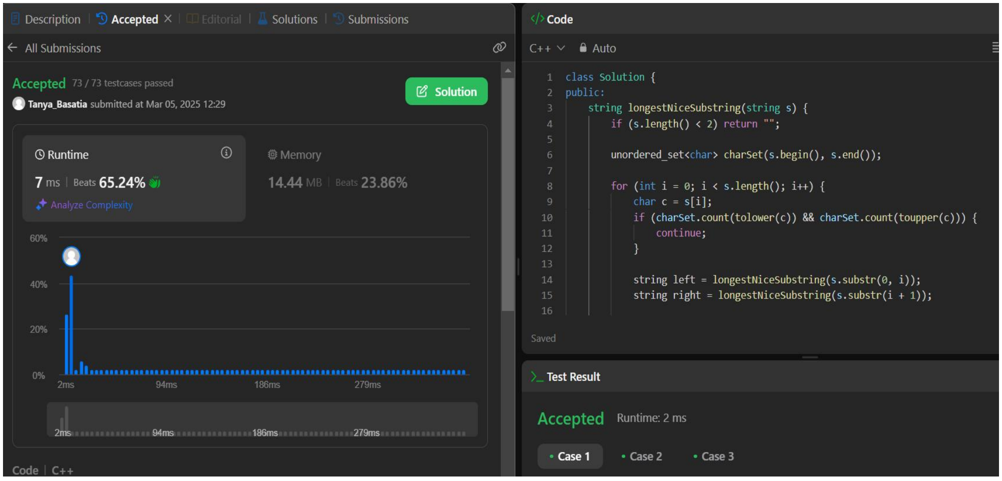
Name- Tanya

UID- 22BCS15446

Section-22BCS_IOT_605-B

1. Longest Nice Substring

```
class Solution {
public:
    string longestNiceSubstring(string s) {
        if (s.length() < 2) return "";

        unordered_set<char> charSet(s.begin(), s.end());

        for (int i = 0; i < s.length(); i++) {
            char c = s[i];
            if (charSet.count(tolower(c)) && charSet.count(toupper(c))) {
                continue;
            }

            string left = longestNiceSubstring(s.substr(0, i));
            string right = longestNiceSubstring(s.substr(i + 1));

            return (left.length() >= right.length()) ? left : right;
        }
        return s;
    }
};
```



Accepted 73 / 73 testcases passed

Tanya_Basatia submitted at Mar 05, 2025 12:29

Runtime: 7 ms | Beats 65.24%

Memory: 14.44 MB | Beats 23.86%

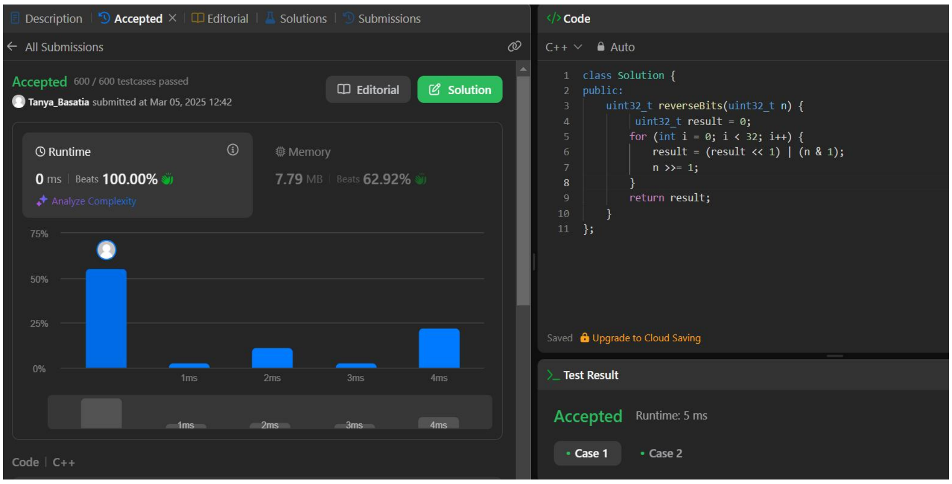
```
class Solution {
public:
    string longestNiceSubstring(string s) {
        if (s.length() < 2) return "";

        unordered_set<char> charSet(s.begin(), s.end());

        for (int i = 0; i < s.length(); i++) {
            char c = s[i];
            if (charSet.count(tolower(c)) && charSet.count(toupper(c))) {
                continue;
            }

            string left = longestNiceSubstring(s.substr(0, i));
            string right = longestNiceSubstring(s.substr(i + 1));

```

Accepted Runtime: 2 ms

2. Reverse Bits

```
class Solution {
public:
  uint32_t reverseBits(uint32_t n) {
     uint32_t result = 0;
    for (int i = 0; i < 32; i++) {
      result = (result << 1) | (n & 1);
      n >>= 1;
    }
```

```
        return result;
    }
};
```



Accepted 600 / 600 testcases passed

Tanya_Basatia submitted at Mar 05, 2025 12:42

Runtime: 0 ms | Beats 100.00%

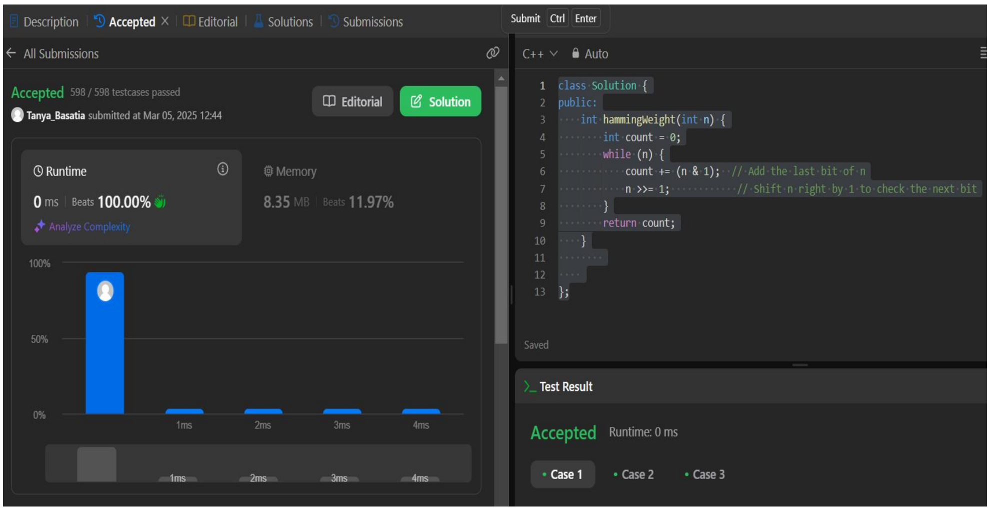
Memory: 7.79 MB | Beats 62.92%

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

Accepted Runtime: 5 ms

3. Number of 1 Bits

```
class Solution {
public:
    int hammingWeight(int n) {
        int count = 0;
        while (n) {
            count += (n & 1);  // Add the last bit of n
            n >>= 1;           // Shift n right by 1 to check the next bit
        }
        return count;
    }

};
```



Accepted 598 / 598 testcases passed

Tanya_Basatia submitted at Mar 05, 2025 12:44

Runtime: 0 ms | Beats 100.00%

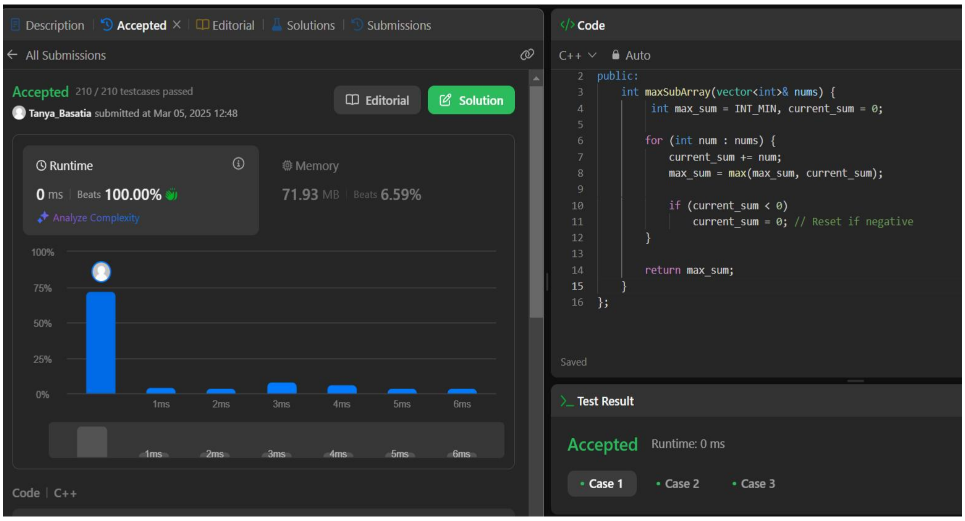
Memory: 8.35 MB | Beats 11.97%

```
class Solution {
public:
    int hammingWeight(int n) {
        int count = 0;
        while (n) {
            count += (n & 1);  // Add the last bit of n
            n >>= 1;           // Shift n right by 1 to check the next bit
        }
        return count;
    }


};
```

Accepted Runtime: 0 ms

4. Maximum Subarray

```
class Solution {
public:
  int maxSubArray(vector<int>& nums) {
     int max_sum = INT_MIN, current_sum = 0;

    for (int num : nums) {
      current_sum += num;
      max_sum = max(max_sum, current_sum);

      if (current_sum < 0)
        current_sum = 0; // Reset if negative
    }

    return max_sum;
  }
};
```



Accepted 210 / 210 testcases passed

Tanya_Basatia submitted at Mar 05, 2025 12:48

Runtime 0 ms | Beats 100.00%

Memory 71.93 MB | Beats 6.59%

Code | C++

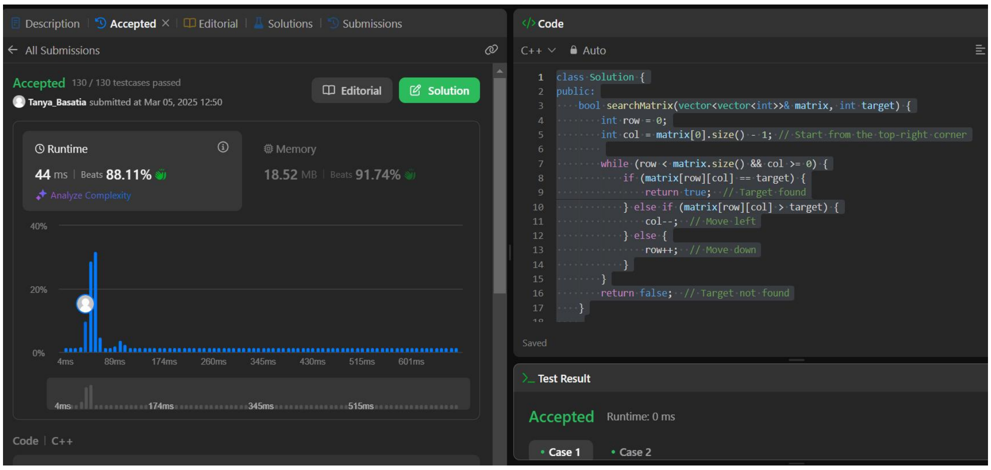
```
public:
    int maxSubArray(vector<int>& nums) {
         int max_sum = INT_MIN, current_sum = 0;

        for (int num : nums) {
            current_sum += num;
            max_sum = max(max_sum, current_sum);

            if (current_sum < 0)
                current_sum = 0; // Reset if negative
        }

        return max_sum;
    }
};
```

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

5. Search a 2D Matrix II

```
class Solution {
public:
  bool searchMatrix(vector<vector<int>>& matrix, int target) {
    int row = 0;
    int col = matrix[0].size() - 1; // Start from the top-right corner
    while (row < matrix.size() && col >= 0) {
      if (matrix[row][col] == target) {
        return true;  // Target found
      } else if (matrix[row][col] > target) {
        col--;  // Move left
      } else {
        row++;  // Move down
      }
    }
    return false;  // Target not found
  } };
```



Accepted 130 / 130 testcases passed

Tanya_Basatia submitted at Mar 05, 2025 12:50

Runtime 44 ms | Beats 88.11%

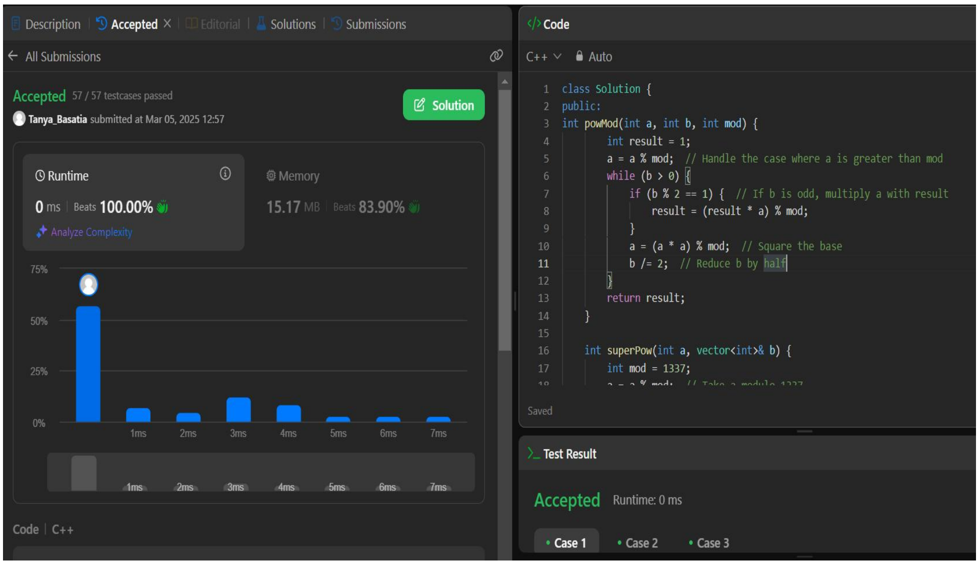
Memory 18.52 MB | Beats 91.74%

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int row = 0;
        int col = matrix[0].size() - 1; // Start from the top-right corner
        
        while (row < matrix.size() && col >= 0) {
            if (matrix[row][col] == target) {
                return true;  // Target found
            } else if (matrix[row][col] > target) {
                col--;  // Move left
            } else {
                row++;  // Move down
            }
        }
        return false;  // Target not found
    }
```

Accepted Runtime: 0 ms

6. Super Pow

```
class Solution {
public:
int powMod(int a, int b, int mod) {
    int result = 1;
    a = a % mod;  // Handle the case where a is greater than mod
    while (b > 0) {
      if (b % 2 == 1) {  // If b is odd, multiply a with result
        result = (result * a) % mod;
      }
      a = (a * a) % mod;  // Square the base
      b /= 2;  // Reduce b by half
    }
    return result;
  }

  int superPow(int a, vector<int>& b) {
    int mod = 1337;
    a = a % mod;  // Take a modulo 1337

    int result = 1;
    // Process each digit in array b
    for (int i = 0; i < b.size(); i++) {
      // Update the result with current digit in b
      result = (powMod(result, 10, mod) * powMod(a, b[i], mod)) % mod;
    }

    return result;
  }

};
```



Accepted 57 / 57 testcases passed

Tanya_Basatia submitted at Mar 05, 2025 12:57

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

15.17 MB | Beats 83.90%

```
class Solution {
public:
int powMod(int a, int b, int mod) {
        int result = 1;
        a = a % mod;  // Handle the case where a is greater than mod
        while (b > 0) {
            if (b % 2 == 1) {  // If b is odd, multiply a with result
                result = (result * a) % mod;
            }
            a = (a * a) % mod;  // Square the base
            b /= 2;  // Reduce b by half
        }
        return result;
    }

    int superPow(int a, vector<int>& b) {
        int mod = 1337;
        a = a % mod;  // Take a modulo 1337
```

Accepted Runtime: 0 ms