Experiment 4

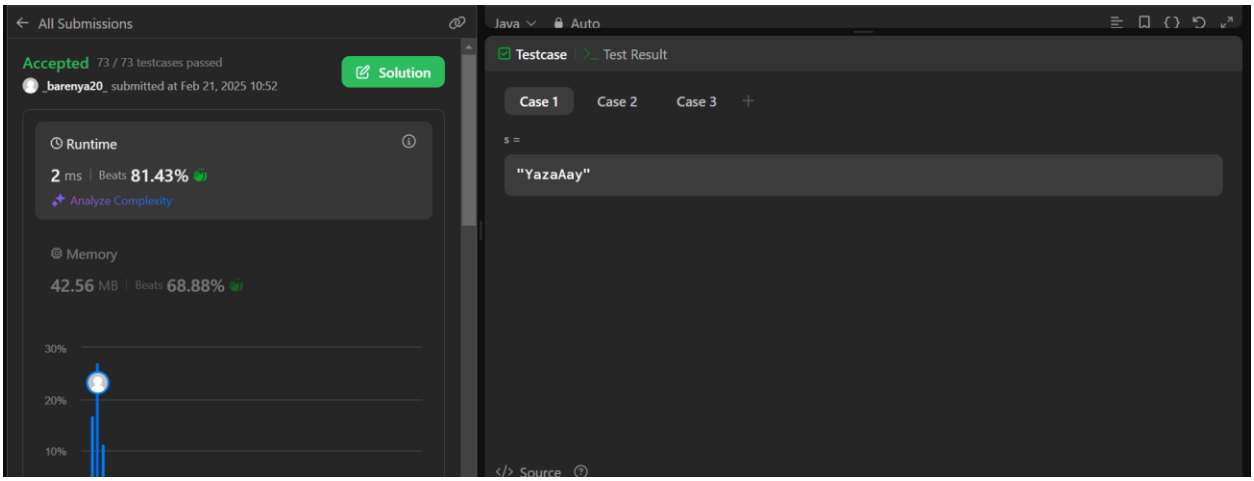
Student Name: Barenya Behera **UID: 22BCS15121**
Branch: BE-CSE **Section/Group:FL_IOT-602/A**
Semester: 6th **Date of Performance: 21/02/25**
Subject Name: Advanced Programming Lab-2 **Subject Code: 22CSP-351**

1. Implementation/Code:

i.

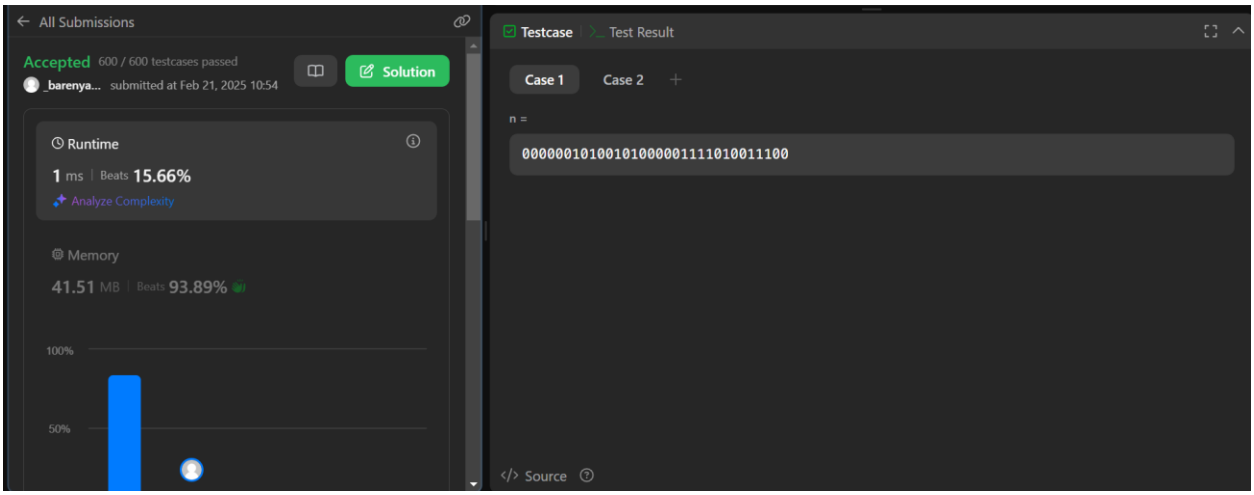
```
class Solution {

public String longestNiceSubstring(String s) {
    if (s.length() < 2) {
        return "";
    }

    Set<Character> set = new HashSet<>();
    for (char c : s.toCharArray()) {
        set.add(c);
    }

    for (int i = 0; i < s.length(); i++) {
        char c = s.charAt(i);
                                if (set.contains(Character.toUpperCase(c)) &&
set.contains(Character.toLowerCase(c))) {
            continue;
        }
        String left = longestNiceSubstring(s.substring(0, i));
```

```
            String right = longestNiceSubstring(s.substring(i + 1));


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



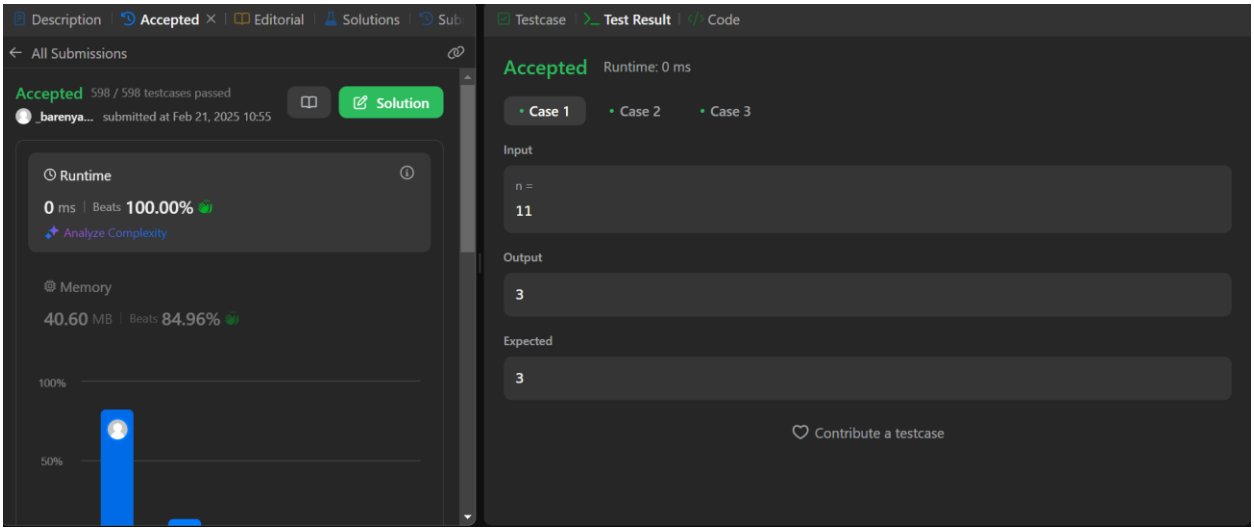
ii.

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



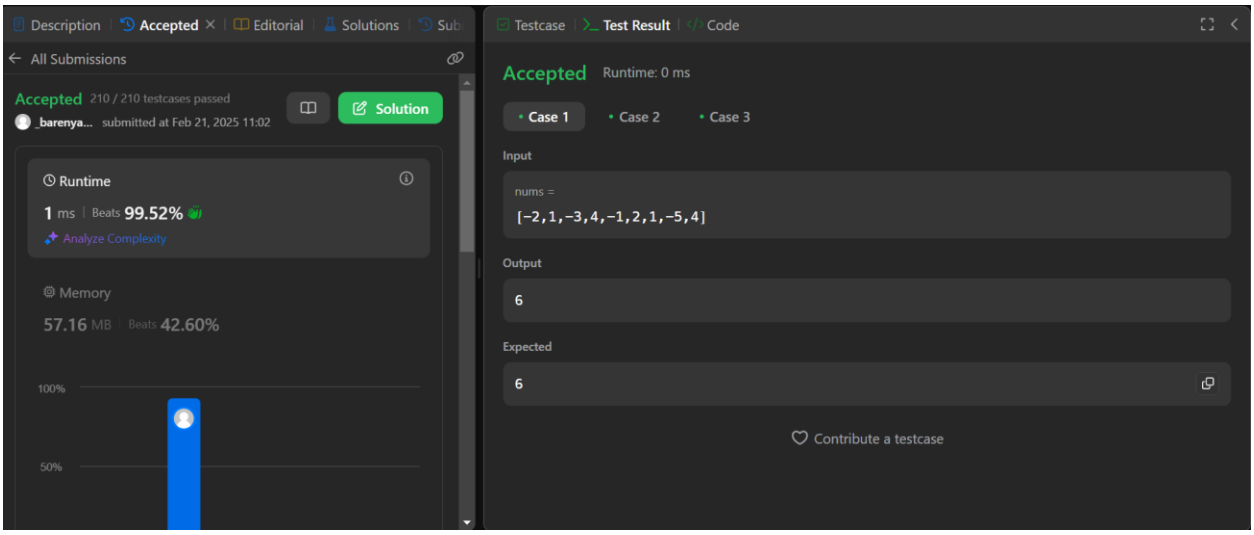
iii.

```
class Solution {
    public int hammingWeight(int n) {
        int count = 0;
        while (n > 0) {
            n = n & (n - 1);
            count++;
        }
        return count;
    }
}
```



iv.

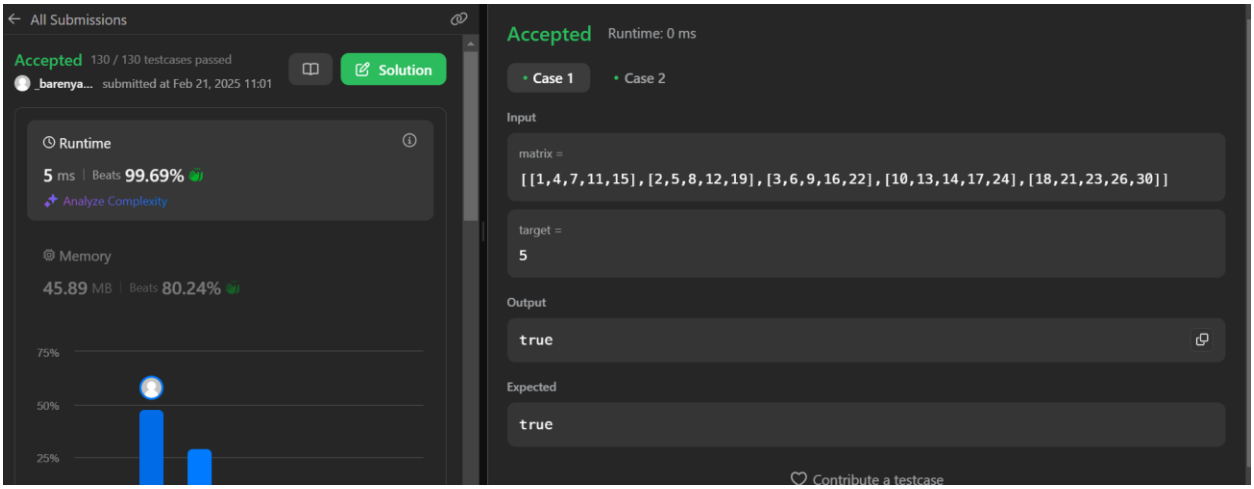
```
class Solution {
  public int maxSubArray(int[] nums) {
     int maxSum=nums[0];
     int currentSum=nums[0];
     for(int i=1;i<nums.length;i++){
        currentSum=Math.max(nums[i],currentSum+nums[i]);
        maxSum=Math.max(maxSum,currentSum);
     }
    return maxSum;
  }
}
```



v.

```
class Solution {
  public boolean searchMatrix(int[][] matrix, int target) {
     if(matrix==null||matrix.length==0||matrix[0].length==0){
        return false;
     }
     int rows=matrix.length;
     int cols=matrix[0].length;
     int row=0,col=cols-1;
```

```
        while(row<rows && col>=0){
            if(matrix[row][col]==target){
                return true;
            }
            else if(matrix[row][col]>target){
                col--;
            }
            else{
                row++;
            }
        }
        return false;
    }
}
```



vi.

```
class Solution {
    private static final int MOD = 1337;
    private int powerMod(int x, int y, int mod) {
        int result = 1;
        x = x % mod;
        while (y > 0) {
            if (y % 2 == 1) {
```

```
                result = (result * x) % mod;
            }
            y /= 2;
            x = (x * x) % mod;
        }
        return result;
    }

    public int superPow(int a, int[] b) {
        a %= MOD;
        int exponent = 0;
        for (int digit : b) {
            exponent = (exponent * 10 + digit) % 1140;
        }
        if (exponent == 0) {
            exponent = 1140;
        }
        return powerMod(a, exponent, MOD);
    }
}
```