DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

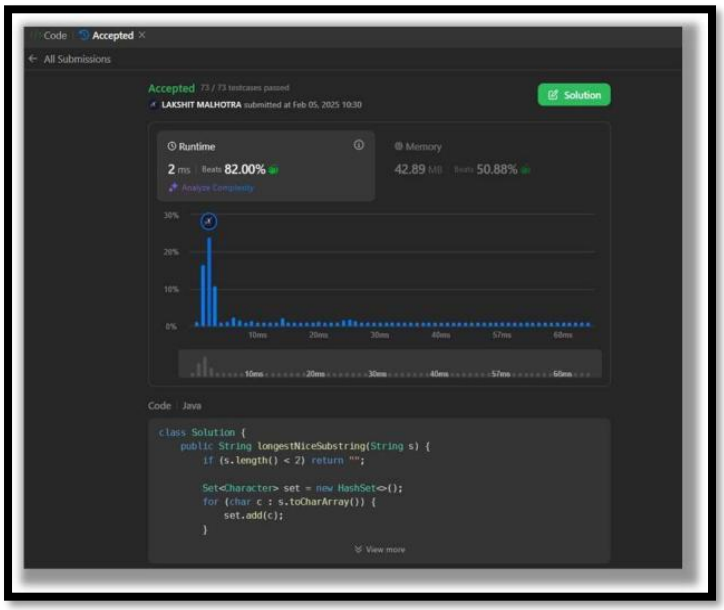
ASSIGNMENT 4

STUDENT NAME: LAKSHIT MALHOTRA **UID:** 22BCS13047
BRANCH: CSE **SECTION:** 22BCS_FL_IOT_601A
SEMESTER: 6 **DATE OF SUBMISSION:** 20/2/25
SUBJECT NAME: AP LAB -2 **SUBJECT CODE:** 22CSP-351

LEET CODE QUESTIONS :

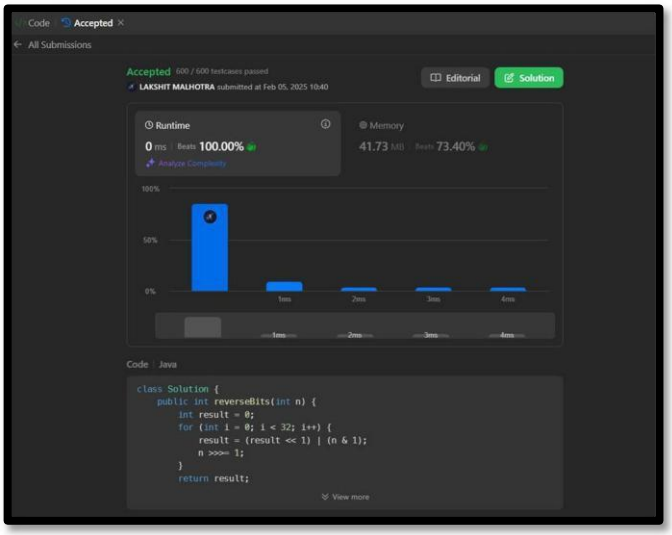
1763.LONGEST NICE SUBSTRING

```
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.toLowerCase(c)) &&
set.contains(Character.toUpperCase(c)))) {
                String left = longestNiceSubstring(s.substring(0, i));
                String right = longestNiceSubstring(s.substring(i + 1));
                return left.length() >= right.length() ? left : right;
            }
        }
        return s;
    }
}
```



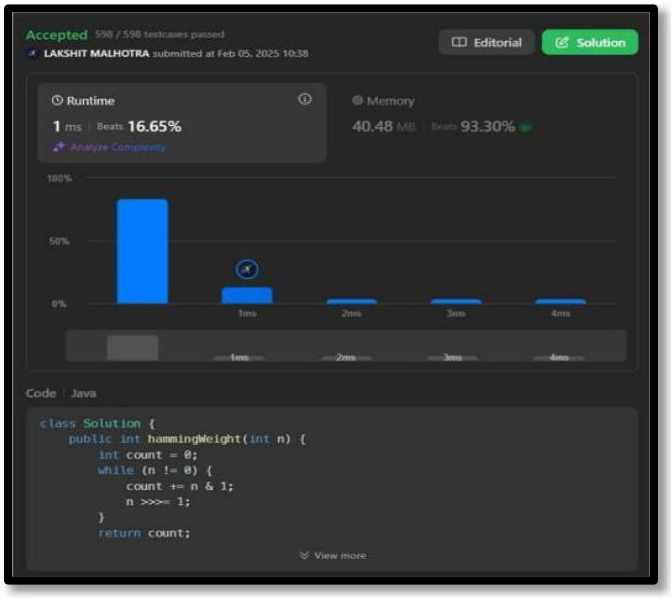
190.REVERSE BITS

```
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;
    }
```



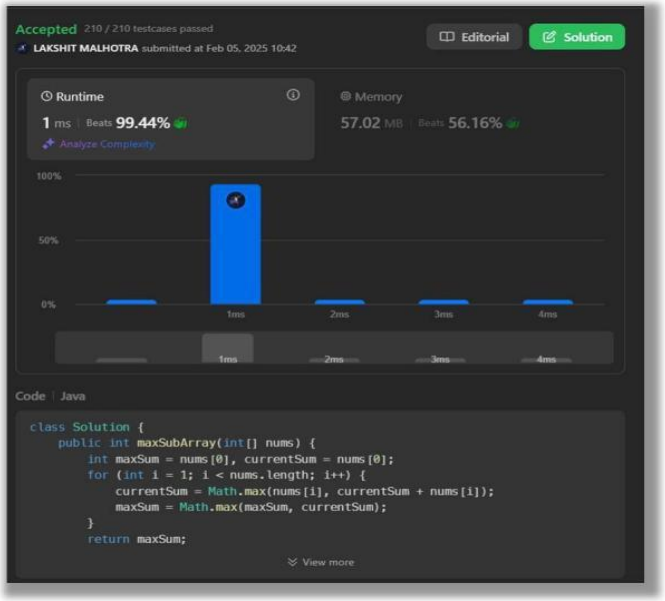
191.NUMBER OF 1 BITS

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



53.MAXIMUM SUBARRAY

```
class Solution {
    public int maxSubArray(int[] nums) {
        int maxSum = nums[0], 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;
    }
}
```



240.SEARCH A 2D MATRIX II

```
class Solution {
    public boolean searchMatrix(int[][] matrix, int target) {
        int m = matrix.length, n = matrix[0].length;
        int row = 0, col = n - 1;
        while (row < m && col >= 0) {
            if (matrix[row][col] == target) return true;
            if (matrix[row][col] > target) col--;
            else row++;
        }
        return false;
    }
}
```

372.SUPER POW

```
class Solution {
    private static final int MOD = 1337;
    public int superPow(int a, int[] b) {
        int result = 1;
        a %= MOD;
        for (int digit : b) {
            result = (pow(result, 10) * pow(a, digit)) % MOD;
        }
        return result;
    }
    private int pow(int x, int n) {
        int result = 1;
        while (n > 0) {
            if ((n & 1) == 1) result = (result * x) % MOD;
            x = (x * x) % MOD;
            n >>= 1;
        }
  return result;
    }
```


Accepted 57 / 57 testcases passed
LAKSHIT MALHOTRA submitted at Feb 05, 2025 10:50
Solution
Runtime
4 ms | Beats 76.00%
Analyze Complexity
Memory
44.57 MB | Beats 38.82%
60%
40%
20%
0%
1ms
6ms
12ms
25ms
32ms
38ms
Code | Java
class Solution {
private static final int MOD = 1337;
public int superPow(int a, int[] b) {
int result = 1;
a %= MOD;
for (int digit : b) {
View more