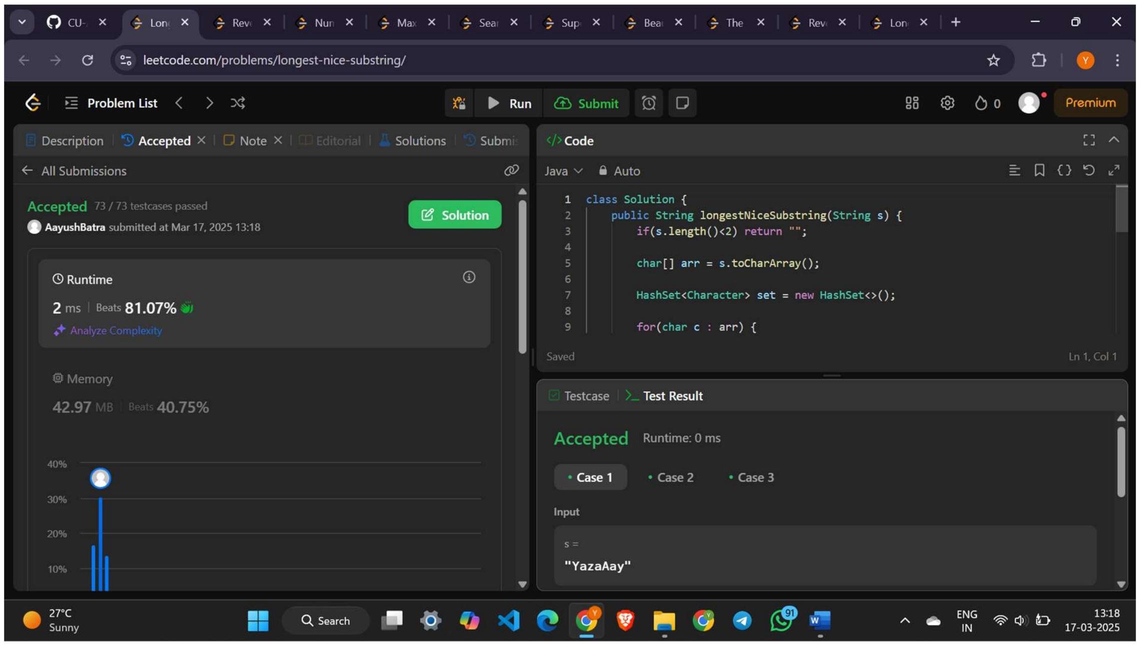
AP Assignment 4

1. Longest Nice Substring: https://leetcode.com/problems/longest-nice-substring/description/



leetcode.com/problems/longest-nice-substring/

Accepted 73 / 73 testcases passed
AayushBatra submitted at Mar 17, 2025 13:18

Runtime
2 ms | Beats 81.07%

Memory
42.97 MB | Beats 40.75%

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

        char[] arr = s.toCharArray();

        HashSet<Character> set = new HashSet<>();

        for(char c : arr) {
```

Test Result

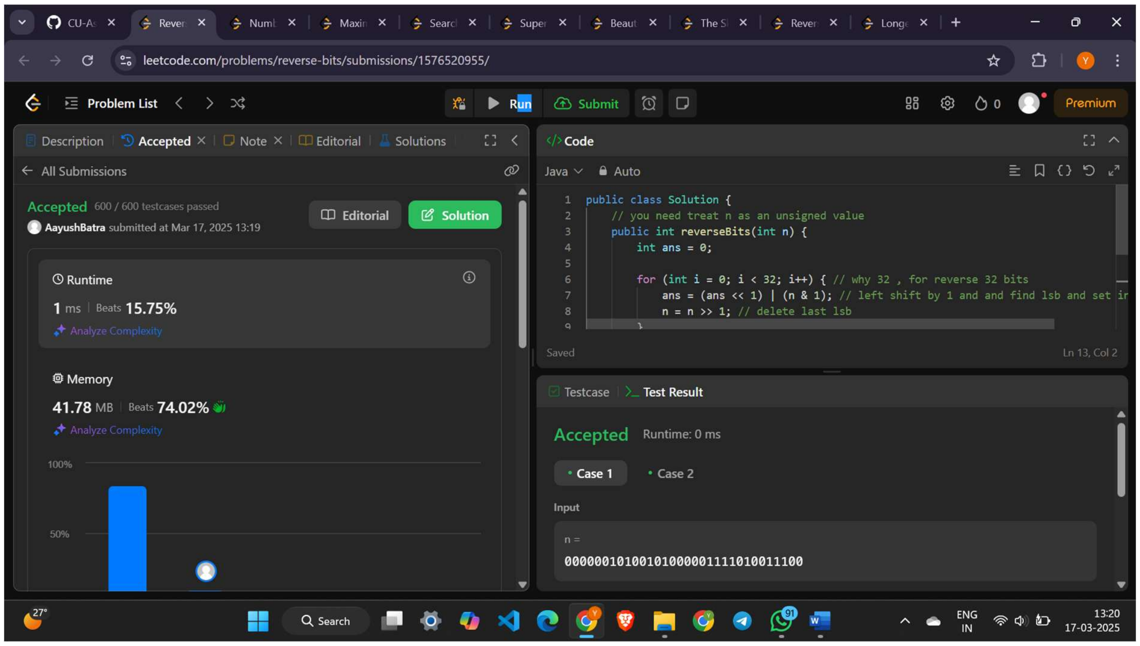
Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3

Input

s =
"YazaAay"

2. Reverse Bits: https://leetcode.com/problems/reverse-bits/description/



leetcode.com/problems/reverse-bits/submissions/1576520955/

Accepted 600 / 600 testcases passed
AayushBatra submitted at Mar 17, 2025 13:19

Runtime
1 ms | Beats 15.75%

Memory
41.78 MB | Beats 74.02%

```java
public class Solution {
    // you need treat n as an unsigned value
    public int reverseBits(int n) {
        int ans = 0;

        for (int i = 0; i < 32; i++) { // why 32 , for reverse 32 bits
            ans = (ans << 1) | (n & 1); // left shift by 1 and and find lsb and set ir
            n = n >> 1; // delete last lsb
```

Test Result

Accepted Runtime: 0 ms

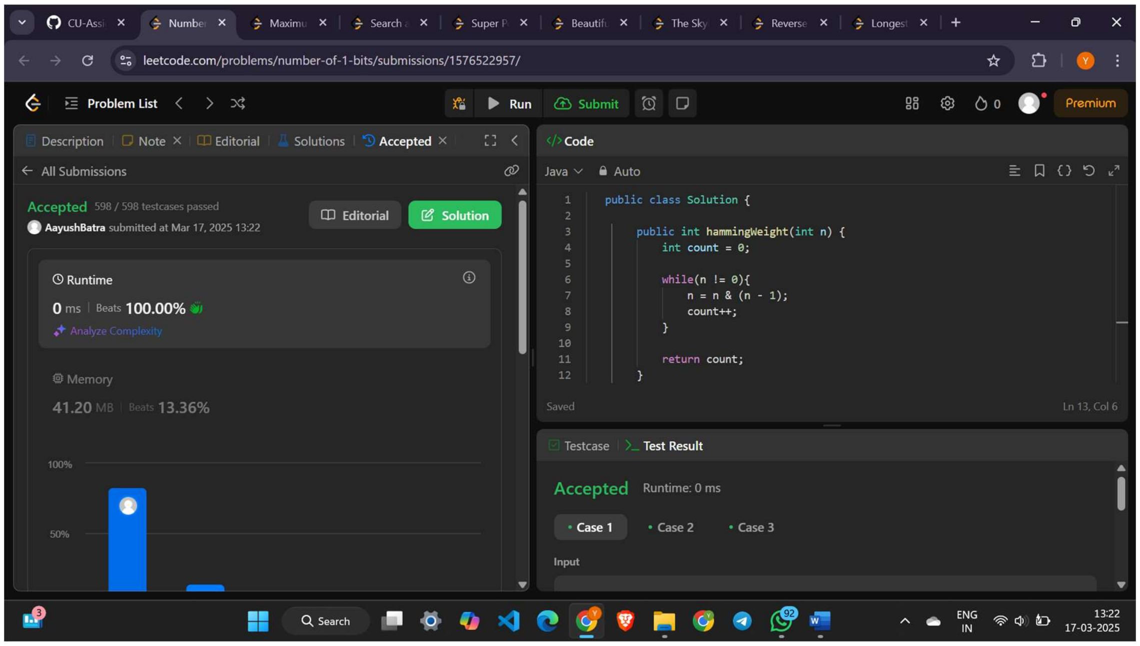
Case 1 · Case 2

Input

n =
00000010100101000001111010011100

3. Number of 1 Bits: https://leetcode.com/problems/number-of-1-bits/description/

AP Assignment 4



Accepted 598 / 598 testcases passed
AayushBatra submitted at Mar 17, 2025 13:22

Runtime
0 ms | Beats 100.00%
Analyze Complexity

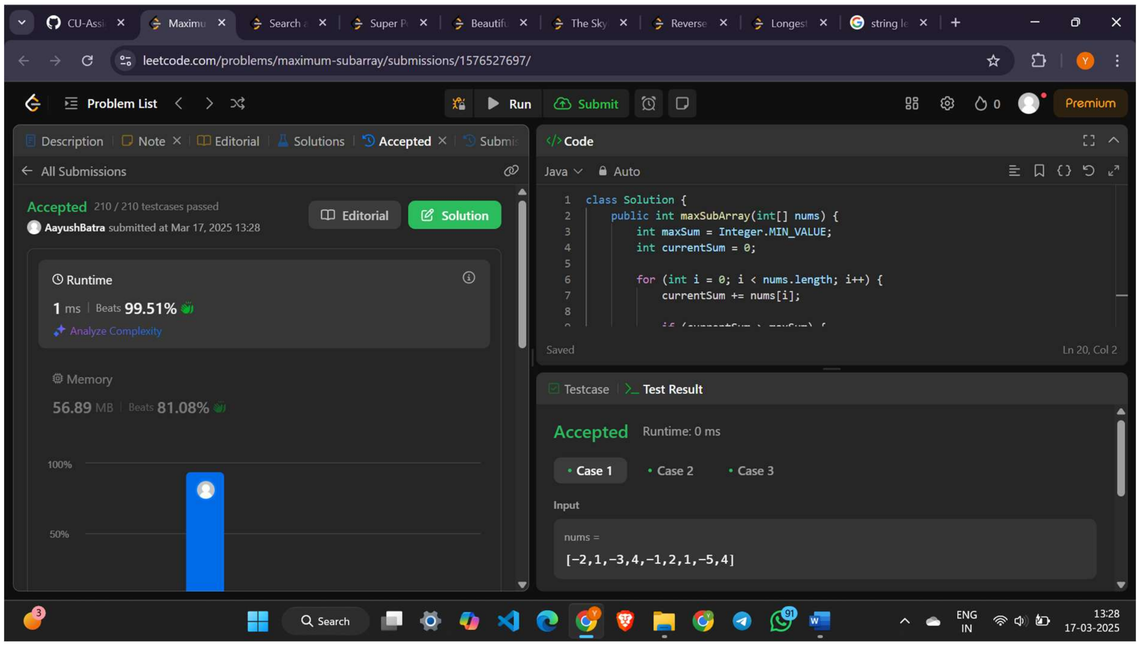
Memory
41.20 MB | Beats 13.36%

```java
public class Solution {

    public int hammingWeight(int n) {
        int count = 0;

        while(n != 0){
            n = n & (n - 1);
            count++;
        }

        return count;
    }
```

Saved — Ln 13, Col 6

Accepted Runtime: 0 ms

4. Maximum Subarray: https://leetcode.com/problems/maximum-subarray/description/



Accepted 210 / 210 testcases passed
AayushBatra submitted at Mar 17, 2025 13:28

Runtime
1 ms | Beats 99.51%
Analyze Complexity

Memory
56.89 MB | Beats 81.08%

```java
class Solution {
    public int maxSubArray(int[] nums) {
        int maxSum = Integer.MIN_VALUE;
        int currentSum = 0;

        for (int i = 0; i < nums.length; i++) {
            currentSum += nums[i];

```

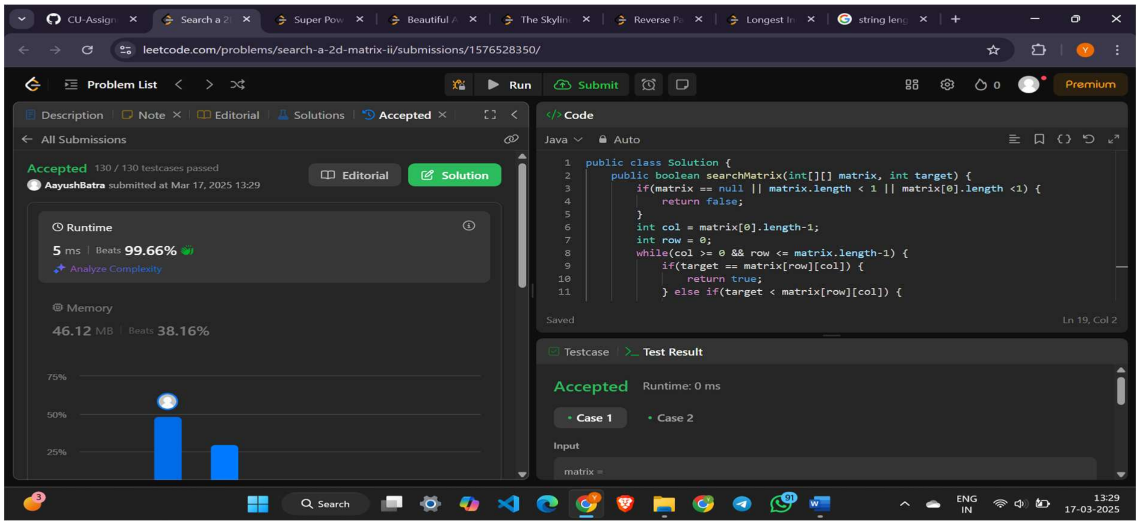
Saved — Ln 20, Col 2

Accepted Runtime: 0 ms

nums =
[-2,1,-3,4,-1,2,1,-5,4]

5. Search a 2D Matrix II: https://leetcode.com/problems/search-a-2d-matrix-ii/description/

AP Assignment 4



leetcode.com/problems/search-a-2d-matrix-ii/submissions/1576528350/

Accepted 130 / 130 testcases passed
AayushBatra submitted at Mar 17, 2025 13:29

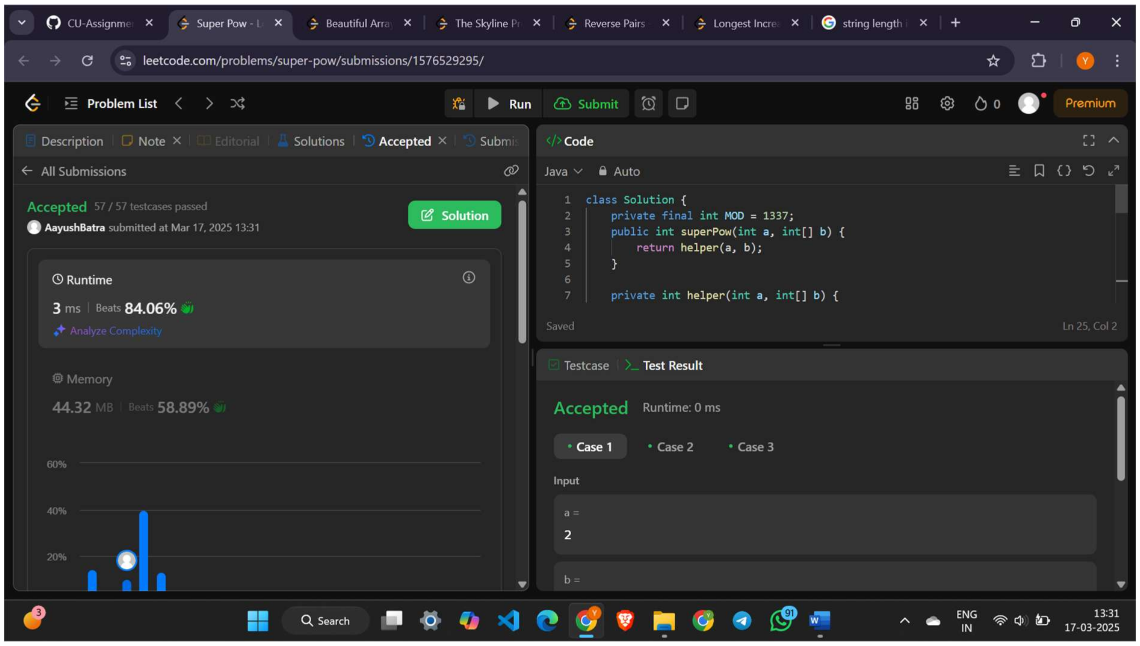
Runtime
5 ms | Beats 99.66%

Memory
46.12 MB | Beats 38.16%

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

Accepted Runtime: 0 ms

6. Super Pow: https://leetcode.com/problems/super-pow/description/



leetcode.com/problems/super-pow/submissions/1576529295/

Accepted 57 / 57 testcases passed
AayushBatra submitted at Mar 17, 2025 13:31

Runtime
3 ms | Beats 84.06%

Memory
44.32 MB | Beats 58.89%

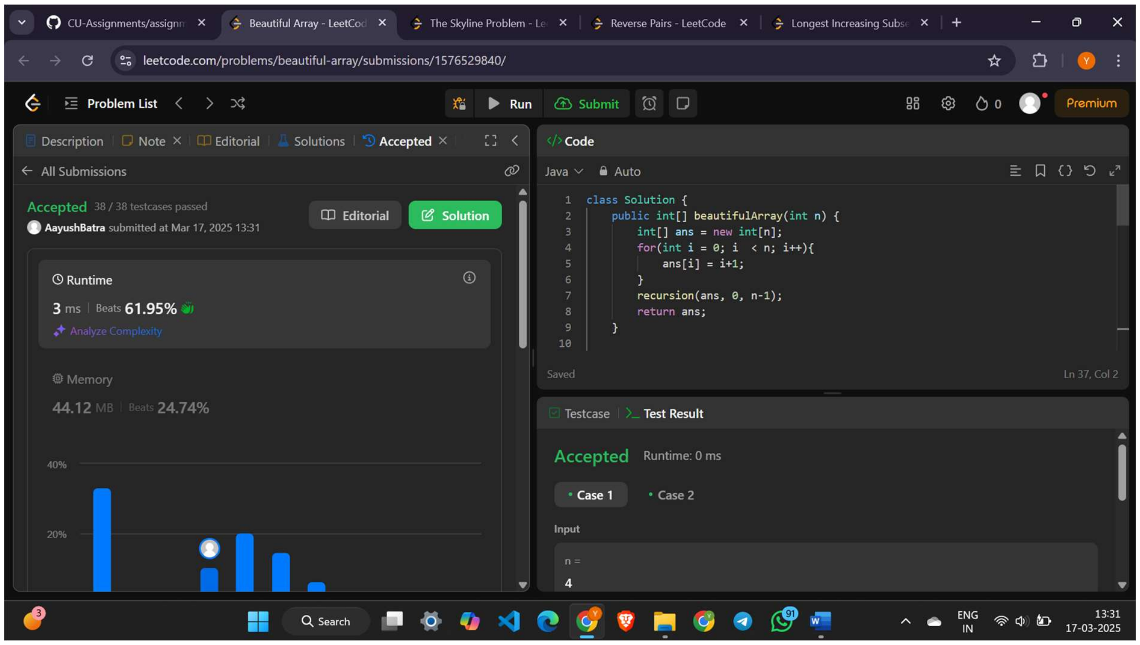
```java
class Solution {
    private final int MOD = 1337;
    public int superPow(int a, int[] b) {
        return helper(a, b);
    }

    private int helper(int a, int[] b) {
```

Accepted Runtime: 0 ms

Input
a =
2
b =

7. Beautiful Array: https://leetcode.com/problems/beautiful-array/description/

AP Assignment 4



leetcode.com/problems/beautiful-array/submissions/1576529840/

Accepted 38 / 38 testcases passed
AayushBatra submitted at Mar 17, 2025 13:31

Runtime
3 ms | Beats 61.95%

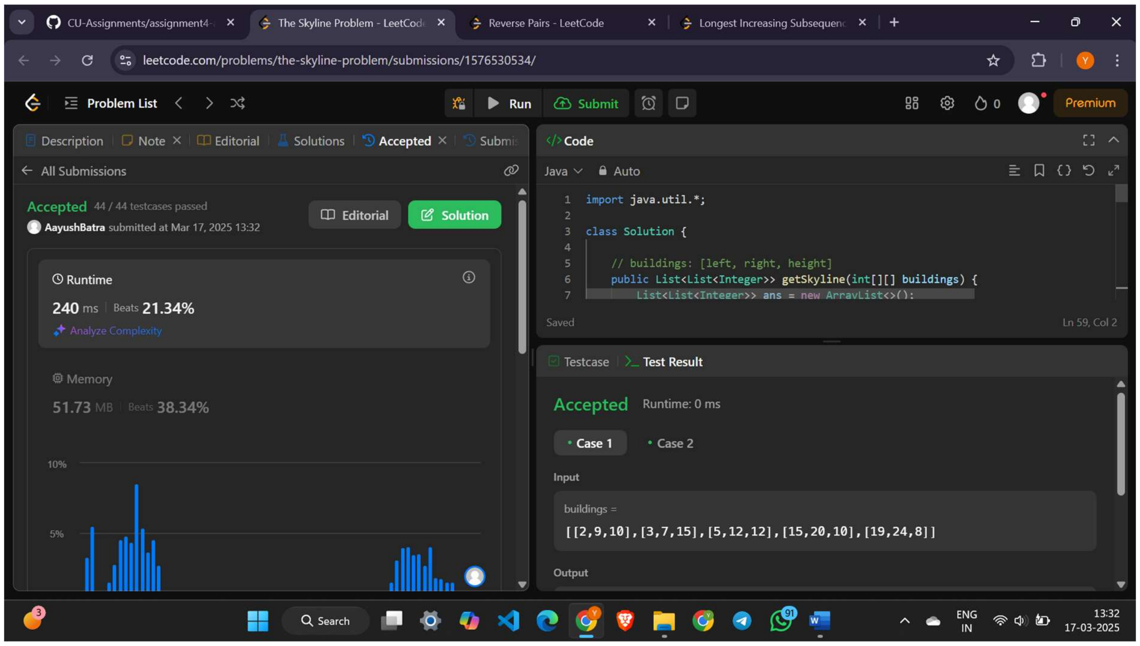
Memory
44.12 MB | Beats 24.74%

```java
class Solution {
    public int[] beautifulArray(int n) {
        int[] ans = new int[n];
        for(int i = 0; i  < n; i++){
            ans[i] = i+1;
        }
        recursion(ans, 0, n-1);
        return ans;
    }
```

Accepted Runtime: 0 ms

Input
n =
4

8. The Skyline Problem: https://leetcode.com/problems/the-skyline-problem/description/



leetcode.com/problems/the-skyline-problem/submissions/1576530534/

Accepted 44 / 44 testcases passed
AayushBatra submitted at Mar 17, 2025 13:32

Runtime
240 ms | Beats 21.34%

Memory
51.73 MB | Beats 38.34%

```java
import java.util.*;

class Solution {

    // buildings: [left, right, height]
    public List<List<Integer>> getSkyline(int[][] buildings) {
        List<List<Integer>> ans = new ArrayList<>();
```

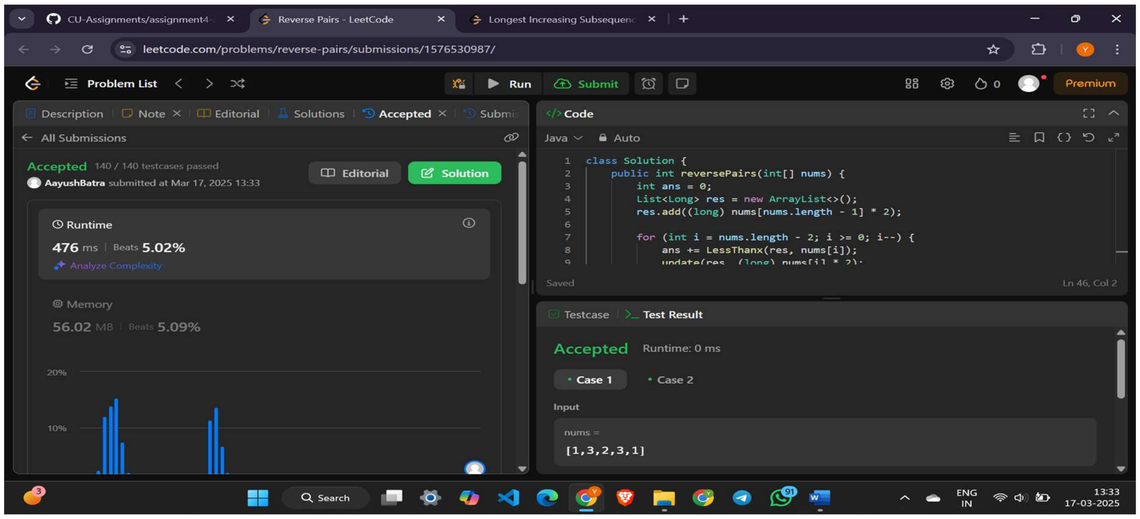
Accepted Runtime: 0 ms

Input
buildings =
[[2,9,10],[3,7,15],[5,12,12],[15,20,10],[19,24,8]]

Output

9. Reverse Pairs: https://leetcode.com/problems/reverse-pairs/description/

AP Assignment 4



leetcode.com/problems/reverse-pairs/submissions/1576530987/

Accepted 140 / 140 testcases passed

AayushBatra submitted at Mar 17, 2025 13:33

Runtime 476 ms | Beats 5.02%

Memory 56.02 MB | Beats 5.09%

```java
class Solution {
    public int reversePairs(int[] nums) {
        int ans = 0;
        List<Long> res = new ArrayList<>();
        res.add((long) nums[nums.length - 1] * 2);

        for (int i = nums.length - 2; i >= 0; i--) {
            ans += LessThanx(res, nums[i]);
            update(res, (long) nums[i] * 2);
```

Accepted Runtime: 0 ms

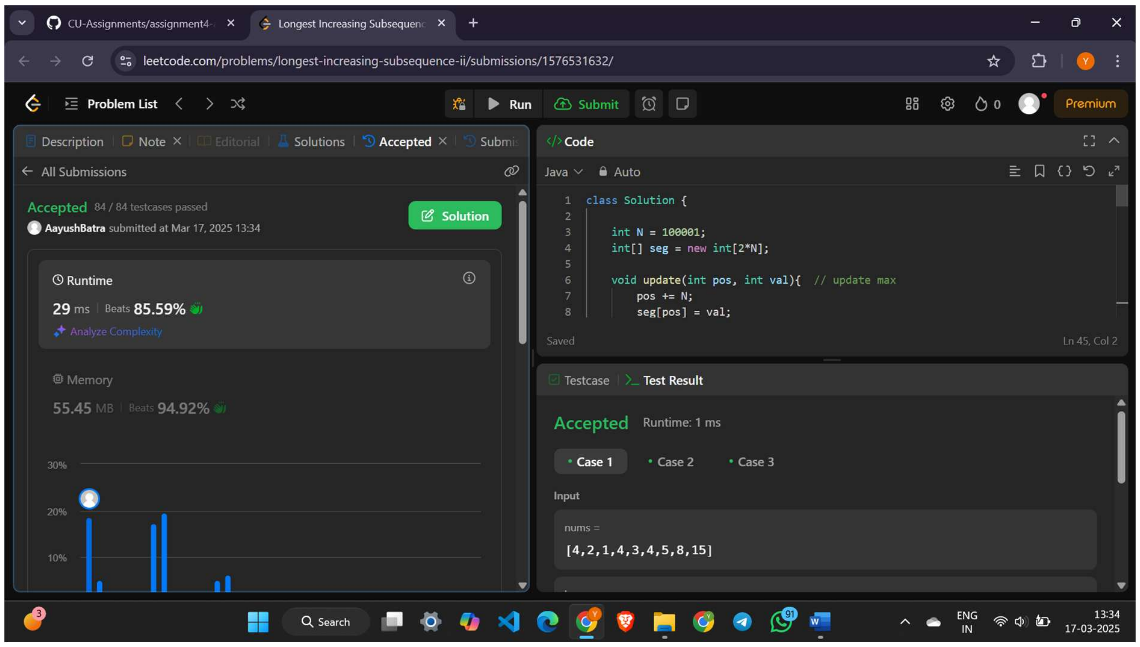
Case 1 Case 2

Input

nums =

```
[1,3,2,3,1]
```

10. Longest Increasing Subsequence II: https://leetcode.com/problems/longest-increasing-subsequence-ii/description/



leetcode.com/problems/longest-increasing-subsequence-ii/submissions/1576531632/

Accepted 84 / 84 testcases passed

AayushBatra submitted at Mar 17, 2025 13:34

Runtime 29 ms | Beats 85.59%

Memory 55.45 MB | Beats 94.92%

```java
class Solution {

    int N = 100001;
    int[] seg = new int[2*N];

    void update(int pos, int val){  // update max
        pos += N;
        seg[pos] = val;
```

Accepted Runtime: 1 ms

Case 1 Case 2 Case 3

Input

nums =

```
[4,2,1,4,3,4,5,8,15]
```