Name: Anshul

UID: 22BCS16477

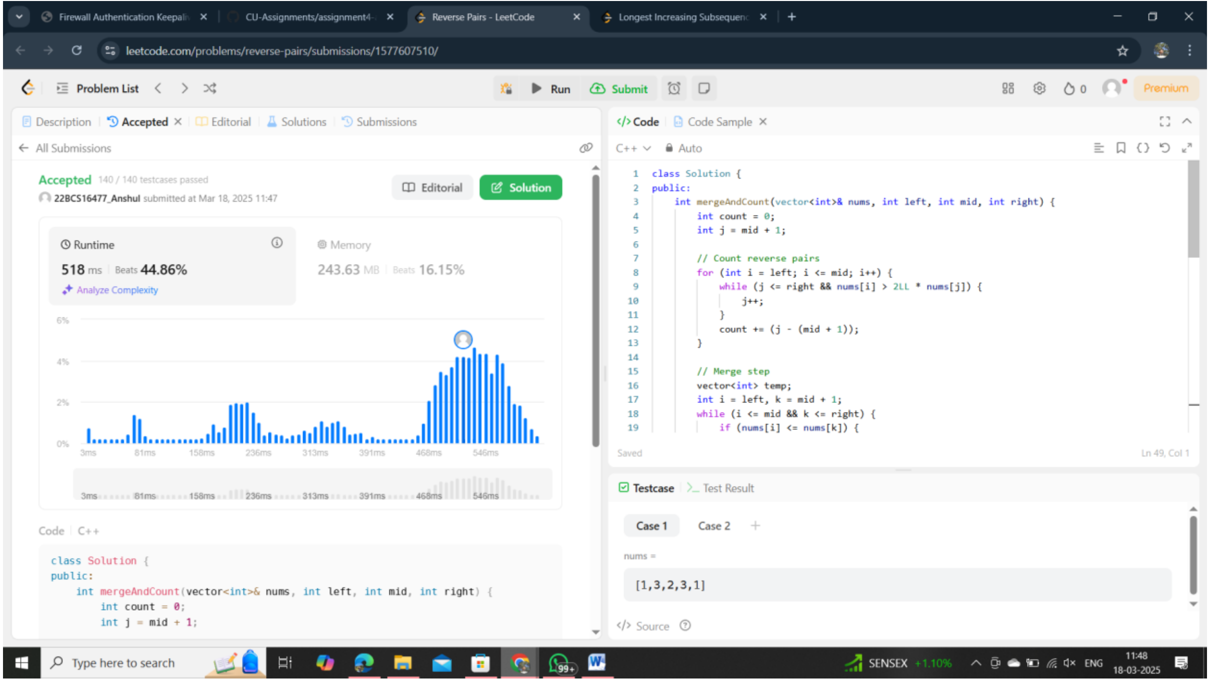
Section/Group: 609(B)

Reverse Pairs

Code:

```
class Solution {
public:
    int mergeAndCount(vector<int>& nums, int left, int mid, int right) {
        int count = 0;
        int j = mid + 1;

        // Count reverse pairs
        for (int i = left; i <= mid; i++) {
            while (j <= right && nums[i] > 2LL * nums[j]) {
                j++;
            }
            count += (j - (mid + 1));
        }

        // Merge step
        vector<int> temp;
        int i = left, k = mid + 1;
        while (i <= mid && k <= right) {
            if (nums[i] <= nums[k]) {
                temp.push_back(nums[i++]);
            } else {
                temp.push_back(nums[k++]);
            }
        }
        while (i <= mid) temp.push_back(nums[i++]);
        while (k <= right) temp.push_back(nums[k++]);

        // Copy sorted values back to original array
        for (int i = left; i <= right; i++) {
            nums[i] = temp[i - left];
        }

        return count;
    }

    int mergeSortAndCount(vector<int>& nums, int left, int right) {
        if (left >= right) return 0;
        int mid = left + (right - left) / 2;
        int count = mergeSortAndCount(nums, left, mid);
        count += mergeSortAndCount(nums, mid + 1, right);
        count += mergeAndCount(nums, left, mid, right);
        return count;
    }

    int reversePairs(vector<int>& nums) {
        return mergeSortAndCount(nums, 0, nums.size() - 1);
    }
};
```

Output:



Accepted 140 / 140 testcases passed

22BCS16477_Anshul submitted at Mar 18, 2025 11:47

518 ms | Beats 44.86%

243.63 MB | Beats 16.15%

Code | C++

```
class Solution {
public:
    int mergeAndCount(vector<int>& nums, int left, int mid, int right) {
        int count = 0;
        int j = mid + 1;
```

```
class Solution {
public:
    int mergeAndCount(vector<int>& nums, int left, int mid, int right) {
        int count = 0;
        int j = mid + 1;

        // Count reverse pairs
        for (int i = left; i <= mid; i++) {
            while (j <= right && nums[i] > 2LL * nums[j]) {
                j++;
            }
            count += (j - (mid + 1));
        }

        // Merge step
        vector<int> temp;
        int i = left, k = mid + 1;
        while (i <= mid && k <= right) {
            if (nums[i] <= nums[k]) {
```

Saved

Ln 49, Col 1

nums =

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