

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:

The screenshot displays the LeetCode submission interface for the 'Reverse Pairs' problem. The submission is accepted, with 140 out of 140 test cases passed. The runtime is 518 ms, which beats 44.86% of other submissions. The memory usage is 243.63 MB, beating 16.15%. A bar chart shows the distribution of runtimes, with a peak around 518 ms. The code is written in C++ and implements a merge sort algorithm to count reverse pairs. The test case [1, 3, 2, 3, 1] is shown with the expected output.

Runtime Performance:

- Runtime: 518 ms (Beats 44.86%)
- Memory: 243.63 MB (Beats 16.15%)

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]) {

```

Testcase:

Case 1: [1, 3, 2, 3, 1]