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

Experiment - 4

Student Name: Ayush Pathania

UID: **22BCS16023**

Branch: **BE - CSE**

Section/Group: **NTPP 602 A**

Semester: **06**

Sub Code: **22CSP-351**

Subject Name: **Advanced Programming Lab - 2**

Date:13/02/2025

Problem - 1

Aim - The aim is to merge two sorted integer arrays, `nums1` and `nums2`, into a single sorted array in non-decreasing order. The solution should utilize the given integers `m` and `n`, representing the number of valid elements in `nums1` and `nums2` respectively, and store the merged result in `nums1`...

Objective - You are provided with two integer arrays, `nums1` and `nums2`, both sorted in non-decreasing order. The length of `nums1` is at least the sum of the lengths of `nums1` and `nums2`, as the unused space in `nums1` is meant to accommodate the elements of `nums2`. The task is to merge the two arrays efficiently, modifying `nums1` in-place, so that the final array remains sorted. The challenge involves using a two-pointer technique or similar algorithmic approach to achieve this in optimal time and space complexity.

Leetcode profile link :

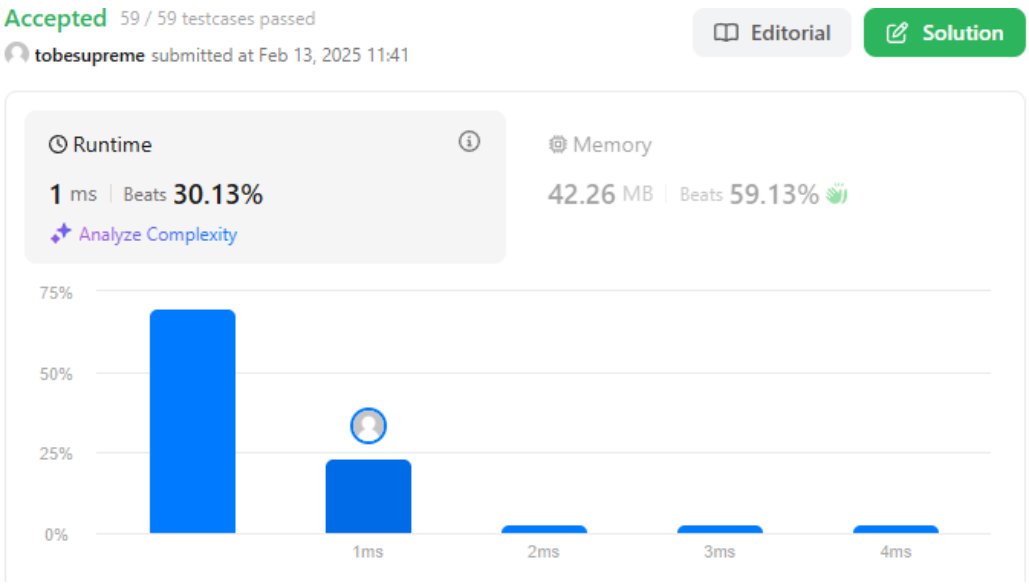
1) https://leetcode.com/problems/merge-sorted-array/submissions/1541372475/

2) https://leetcode.com/problems/sort-colors/submissions/1541389433/

Code -

```
class Solution {
    public void merge(int[] nums1, int m, int[] nums2, int n) {
        for (int j = 0, i = m; j < n; j++) {
            nums1[i] = nums2[j];
            i++;
        }
        Arrays.sort(nums1);
    }
}
```

Output -



Time Complexity : O(n)(Log(N)) **Space Complexity -** O(1)

Problem - 2

Aim - The aim is to sort an array of integers representing colors (red, white, and blue) in-place, where 0, 1, and 2 represent red, white, and blue, respectively, ensuring that the colors are arranged in the order red, white, and blue.

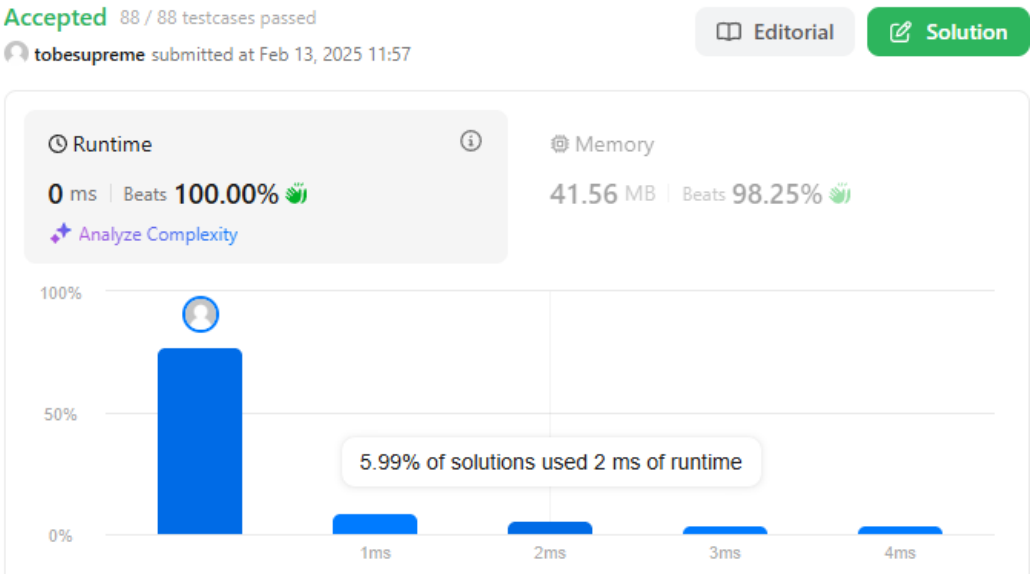
Objective - The objective is to efficiently sort the array without using built-in sorting functions, employing an optimal algorithm, such as the Dutch National Flag algorithm, to achieve the desired arrangement with minimal space and time complexity.

Code -

```
class Solution {
    public void sortColors(int[] nums) {
        //Brute Force Method : Sorting

        Arrays.sort(nums);
        //System.out.println(Arrays.toString(nums));
    }
}
```

Output -



Time Complexity:- O(N log N). **Space Complexity:-** O(1)

Learning Outcomes -

1. **Understanding Data Structures:**
   Learn to manipulate arrays efficiently by understanding how to work with pointers or indices for sorting and merging operations.
2. **Problem-Solving Skills:**
   Enhance problem-solving skills through the application of specific algorithms for merging arrays and sorting based on custom constraints.
3. **Algorithmic Thinking:**
   Develop algorithmic thinking by selecting and implementing efficient algorithms, such as bitwise operations or the Dutch National Flag algorithm for sorting.