

Assignment 5

Name: Abhigyan	Uid: 22BCS10097
Branch: BE_CSE	Semester: 6th
Section: IOT_637-B	Subject: AP Lab II

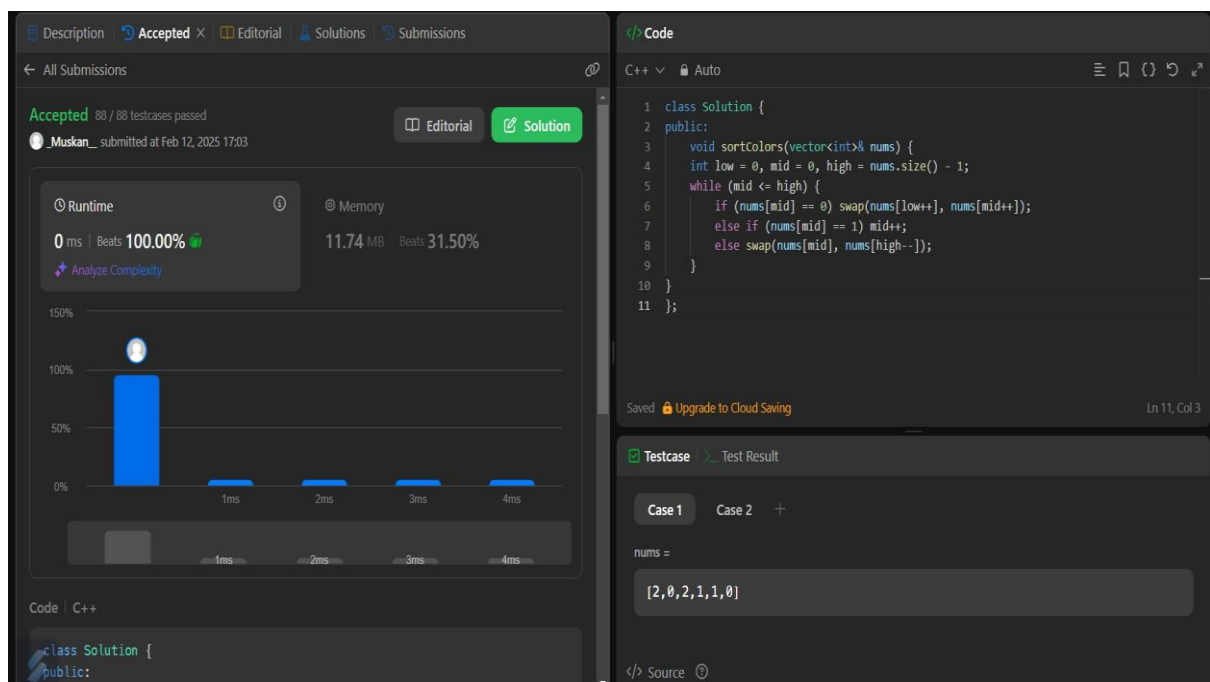
75. Sort Colors

Aim: Given an array nums with n objects colored red, white, or blue, sort them in-place so that objects of the same color are adjacent, with the colors in the order red, white, and blue. We will use the integers 0, 1, and 2 to represent the color red, white, and blue, respectively. You must solve this problem without using the library's sort function.

Code:

```
class Solution {
public:
    void sortColors(vector<int>& nums) {
        int low = 0, mid = 0, high = nums.size() - 1;
        while (mid <= high) {
            if (nums[mid] == 0)
                swap(nums[low++], nums[mid++]);
            else if (nums[mid] == 1)
                mid++;
            else
                swap(nums[mid], nums[high--]);
        }
    }
};
```

Output:



215. Kth Largest Element in an Array

Aim: Given an integer array `nums` and an integer `k`, return the `k`th largest element in the array.

Note that it is the `k`th largest element in the sorted order, not the `k`th distinct element. Can you solve it without sorting?

Code:

```
class Solution {
public:
    int findKthLargest(vector<int>& nums, int k) {
        priority_queue<int, vector<int>, greater<int>>> pq;
        for (int num : nums) {
            pq.push(num);
            if (pq.size() > k)
                pq.pop();
        }
        return pq.top();
    }
};
```

};

Output:

The screenshot displays a LeetCode submission interface. The top navigation bar includes links for Description, Accepted (42 / 42 testcases passed), Editorial, Solutions, and Submissions. The submission is by user **Muskan_**, submitted on Feb 12, 2025 at 17:05. The submission status is **Accepted**.

Runtime: 41 ms | Beats 23.38%
Memory: 61.53 MB | Beats 44.61%

A histogram shows the distribution of runtime times, with a peak around 35ms. The x-axis ranges from 1ms to 239ms, and the y-axis shows the percentage of submissions from 0% to 20%.

Code: C++
The code implements a solution to find the kth largest element in an array. It uses a priority queue (min-heap) to store the first k largest elements. The algorithm iterates through the array, pushing elements into the heap and popping the smallest element if the heap size exceeds k. Finally, the top element of the heap is returned.

```
class Solution {
public:
    int findKthLargest(vector<int>& nums, int k) {
        priority_queue<int, vector<int>, greater<int>> pq;
        for (int num : nums) {
            pq.push(num);
            if (pq.size() > k) pq.pop();
        }
        return pq.top();
    }
};
```

Testcase: Case 1 | Case 2 | +
nums = [3, 2, 1, 5, 6, 4]
k = 2