



Assignment 5.

Student Name: Mannat Gupta

Branch: BE-CSE

Semester: 6th

Subject Name: Advanced Programming Lab-2

UID: 22BCS15281

Section/Group: 608-B

Date of Performance: 03/04/25

Subject Code: 22CSP-351

Aim: Sorting and Searching:

1. **Median of Two Sorted Arrays:** <https://leetcode.com/problems/median-of-two-sorted-arrays/>

```
class Solution {
public:
    double findMedianSortedArrays(vector<int>& nums1, vector<int>&
nums2) {
        if (nums1.size() > nums2.size()) {
            swap(nums1, nums2);
        }

        int m = nums1.size(), n = nums2.size();
        int left = 0, right = m;
        while (left <= right) {
            int partition1 = left + (right - left) / 2;
            int partition2 = (m + n + 1) / 2 - partition1;

            int maxLeft1 = (partition1 == 0) ? INT_MIN : nums1[partition1 - 1];
            int minRight1 = (partition1 == m) ? INT_MAX : nums1[partition1];

            int maxLeft2 = (partition2 == 0) ? INT_MIN : nums2[partition2 - 1];
            int minRight2 = (partition2 == n) ? INT_MAX : nums2[partition2];

            if (maxLeft1 <= minRight2 && maxLeft2 <= minRight1) {
                if ((m + n) % 2 == 0) {
                    return (max(maxLeft1, maxLeft2) + min(minRight1, minRight2))
/ 2.0;
                } else {
                    return max(maxLeft1, maxLeft2);
                }
            }
        }
    }
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
    }  
    } else if (maxLeft1 > minRight2) {  
        right = partition1 - 1;  
    } else {  
        left = partition1 + 1;  
    }  
}  
  
return 0.0;  
}  
};
```

Accepted 2096 / 2096 testcases passed

Mannat Gupta submitted at Apr 03, 2025 18:19

Editorial

Solution

Runtime



0 ms | Beats 100.00%

[Analyze Complexity](#)

Memory

95.12 MB | Beats 63.38%





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

2. Find Peak Element: <https://leetcode.com/problems/find-peak-element/>

```
class Solution {
public:
    int findPeakElement(vector<int>& nums) {
        int left = 0, right = nums.size() - 1;

        while (left < right) {
            int mid = left + (right - left) / 2;
            if (nums[mid] > nums[mid + 1]) {
                right = mid; // Move left
            } else {
                left = mid + 1; // Move right
            }
        }

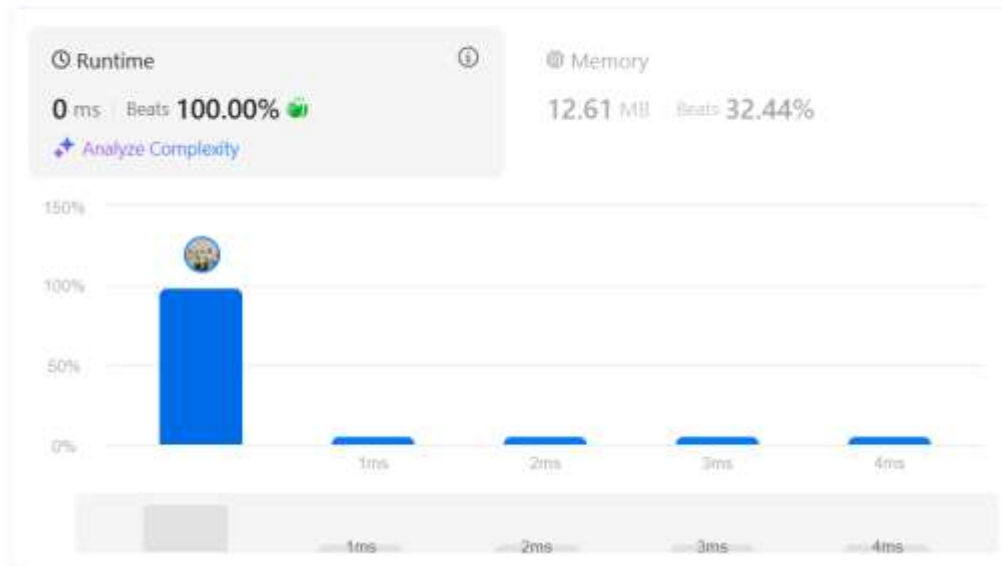
        return left; // Peak element index
    }
};
```

Accepted 68 / 68 testcases passed

Mannat Gupta submitted at Apr 03, 2025 18:18

Editorial

Solution





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

3. First Bad Version: <https://leetcode.com/problems/first-bad-version/>

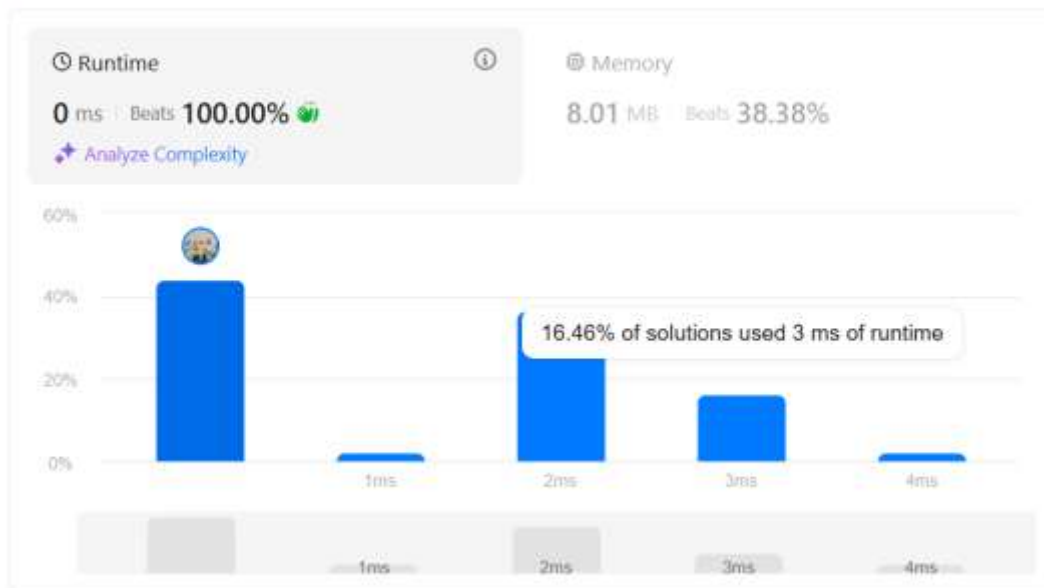
```
class Solution {  
public:  
    int firstBadVersion(int n) {  
        int left = 1, right = n;  
        while (left < right) {  
            int mid = left + (right - left) / 2;  
            if (isBadVersion(mid)) {  
                right = mid; // Search left half  
            } else {  
                left = mid + 1; // Search right half  
            }  
        }  
        return left; // First bad version  
    }  
};
```

Accepted 24 / 24 testcases passed

Mannat Gupta submitted at Apr 03, 2025 18:15

Editorial

Solution





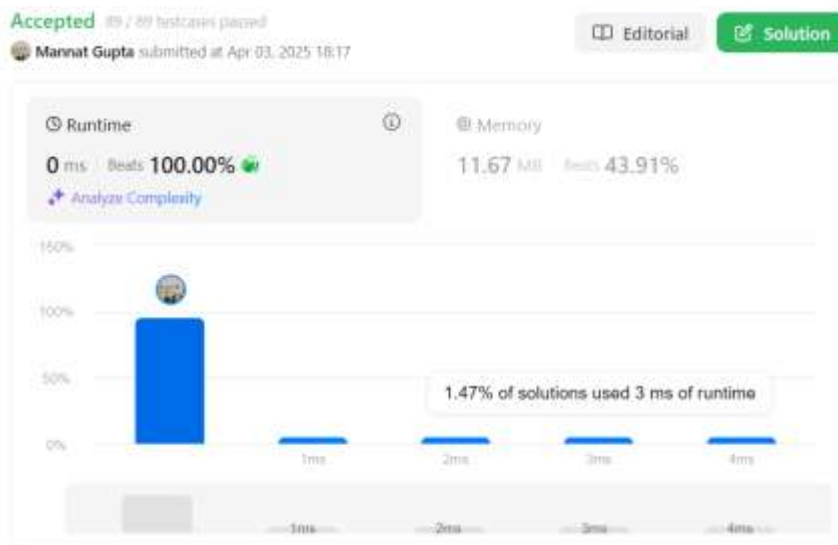
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

4. Sort Colors: <https://leetcode.com/problems/sort-colors/>

```
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 { // nums[mid] == 2
                swap(nums[mid], nums[high--]);
            }
        }
    }
};
```





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

5. Merge Sorted Array: <https://leetcode.com/problems/merge-sorted-array/>

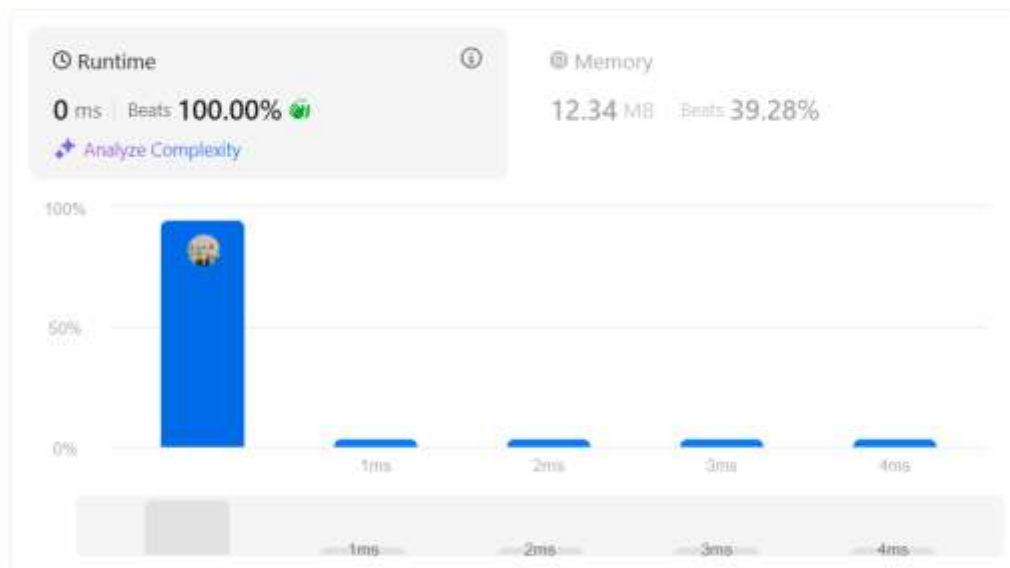
```
class Solution {
public:
    void merge(vector<int>& nums1, int m, vector<int>& nums2, int n) {
        int i = m - 1, j = n - 1, k = m + n - 1;
        while (i >= 0 && j >= 0) {
            if (nums1[i] > nums2[j]) {
                nums1[k--] = nums1[i--];
            } else {
                nums1[k--] = nums2[j--];
            }
        }
        while (j >= 0) {
            nums1[k--] = nums2[j--];
        }
    }
};
```

Accepted 59 / 59 testcases passed

Mannat Gupta submitted at Apr 03, 2025 18:13

Editorial

Solution





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.