

ASSIGNMENT - 5

Student Name: Manreet Kaur

UID: 22BCS15550

Branch: BE-CSE

Section/Group: 608/B

Semester: 6th

Subject Name: AP LAB

1. Merge Sorted Array

The screenshot displays a coding platform interface. At the top, there's a navigation bar with 'Problem List', navigation arrows, and a 'Run' button. Below this is a tabbed interface with 'Description', 'Editorial', 'Solutions', and 'Submissions'. The 'Submissions' tab is active, showing a table with columns: Status, Language, Runtime, Memory, and Notes. A single submission is listed with status 'Accepted', language 'C++', runtime '0 ms', and memory '12.3 MB', dated 'Feb 27, 2025'. Below the table is a 'Code' section showing the C++ implementation of the merge sort algorithm. The code defines a 'Solution' class with a 'merge' function that merges two sorted arrays into a third. The right sidebar shows the user profile 'Manreet_05' with a premium subscription notice and icons for 'My Lists', 'Notebook', 'Submissions', 'Progress', and 'Points'.

```
1 class Solution {
2 public:
3 void merge(vector<int>& nums1, int m, vector<int>& nums2, int n) {
4     int i = m - 1;
5     int j = n - 1;
6     int k = m + n - 1;
7
8     while (i >= 0 && j >= 0) {
9         if (nums1[i] > nums2[j])
10             nums1[k--] = nums1[i--];
11         else
12             nums1[k--] = nums2[j--];
13     }
14
15     while (j >= 0)
16         nums1[k--] = nums2[j--];
17 }
18 };
```


2. First Bad Version

The screenshot displays the same coding platform interface as above, but for the 'First Bad Version' problem. The submission table shows a single submission with status 'Accepted', language 'C++', runtime '2 ms', and memory '7.9 MB', dated '13 minutes ago'. The 'Code' section and the right sidebar are not visible in this screenshot.


</> Code

C++ ▾ 🔒 Auto


```
3 class Solution {
4 public:
5     int firstBadVersion(int n) {
6         int left = 1, right = n;
7         while (left < right) {
8             int mid = left + (right - left) / 2;
9             if (isBadVersion(mid)) {
10                 right = mid;
11             } else {
12                 left = mid + 1;
13             }
14         }
15         return left;
16     }
17 };
```




Manreet_05
Access all features with our Premium subscription!




My Lists




Notebook



Submissions








Progress





Points

Try New Features

3. Sort Colors

  Problem List   

  Run


Description | Editorial | Solutions | Submissions

	Status ▾	Language ▾	Runtime	Memory	Notes
2	Accepted Mar 21, 2025	C++	0 ms	11.7 MB	
1	Accepted Mar 21, 2025	C++	4 ms	11.7 MB	


</> Code | Accepted ✕

C++ ▾ 🔒 Auto


```
1 class Solution {
2 public:
3     void sortColors(vector<int>& nums) {
4         int count0 = 0, count1 = 0, count2 = 0;
5         for (int num : nums) {
6             if (num == 0) count0++;
7             else if (num == 1) count1++;
8             else count2++;
9         }
10        int index = 0;
11        while (count0--) nums[index++] = 0;
12        while (count1--) nums[index++] = 1;
13        while (count2--) nums[index++] = 2;
14    }
15 };
```




Manreet_05
Access all features with our Premium subscription!




My Lists




Notebook



Submissions



Progress



Points

Try New Features

4. Find Peak Element

The screenshot shows a code editor interface for a problem titled "Find Peak Element". The top navigation bar includes "Problem List", "Description", "Editorial", "Solutions", and "Submissions". A status bar at the top right shows "Run". Below the navigation bar, a table displays submission details:

Status	Language	Runtime	Memory	Notes
1 Accepted 11 minutes ago	C++	0 ms	12.5 MB	

The main code editor area shows the following C++ code:

```
1 class Solution {
2 public:
3     int findPeakElement(vector<int>& nums) {
4         int left = 0, right = nums.size() - 1;
5         while (left < right) {
6             int mid = left + (right - left) / 2;
7             if (nums[mid] > nums[mid + 1]) {
8                 right = mid;
9             } else {
10                left = mid + 1;
11            }
12        }
13        return left;
14    }
15};
```



On the right side, a user profile for "Manreet_05" is displayed, along with buttons for "My Lists", "Notebook", "Submissions", "Progress", and "Points". A "Try New Features" button is also visible at the bottom.

5. Median of Two Sorted Arrays

The screenshot shows a code editor interface for a problem titled "Median of Two Sorted Arrays". The top navigation bar includes "Problem List", "Description", "Editorial", "Solutions", and "Submissions". A status bar at the top right shows "Run". Below the navigation bar, a table displays submission details:

Status	Language	Runtime	Memory	Notes
1 Accepted 7 minutes ago	C++	0 ms	95.2 MB	

</> Code

C++   Auto

```
1 class Solution {
2 public:
3     double findMedianSortedArrays(vector<int>& nums1, vector<int>& nums2) {
4         if (nums1.size() > nums2.size()) swap(nums1, nums2);
5         int m = nums1.size(), n = nums2.size();
6         int left = 0, right = m;
7         while (left <= right) {
8             int cut1 = (left + right) / 2;
9             int cut2 = (m + n + 1) / 2 - cut1;
10
11             int maxLeft1, minRight1, maxLeft2, minRight2;
12
13             if (cut1 == 0) maxLeft1 = INT_MIN;
14             else maxLeft1 = nums1[cut1 - 1];
15
16             if (cut1 == m) minRight1 = INT_MAX;
17             else minRight1 = nums1[cut1];
18
19             if (cut2 == 0) maxLeft2 = INT_MIN;
20             else maxLeft2 = nums2[cut2 - 1];
21
22             if (cut2 == n) minRight2 = INT_MAX;
23             else minRight2 = nums2[cut2];
```



Manreet_05

Access all features with our
Premium subscription!



My Lists



Notebook



Submissions



Progress



Points



Try New Features



Orders



My Playgrounds



Settings



Classic Mode