

Course Name:	Competitive Programming Lab	Semester:	IV
Date of Performance:	12/ 01/ 2026	DIV/ Batch No:	B2
Student Name:	Ashwera Hasan	Roll No:	16010124107

Experiment No: 0

Title: Arrays - Searching, Frequency Counting, and Basic Array Analysis

Aim and Objective of the Experiment:

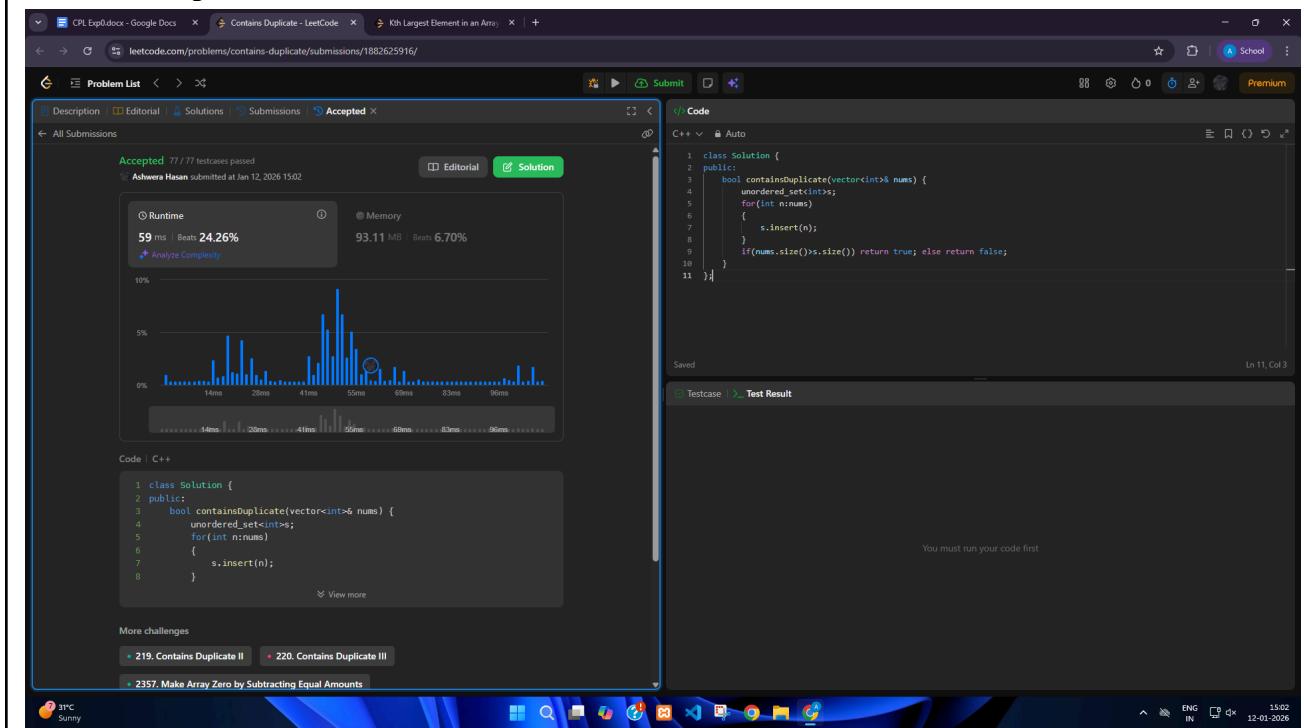
Easy	Count distinct elements in array	LC 217 – Contains Duplicate
Easy	Find first repeating element	LC 2351 – First Letter to Appear Twice
Medium	Minimum difference between any two elements	LC 1200 – Minimum Absolute Difference

COs to be achieved:

Applying various problem-solving paradigms, enabling them to create and implement efficient algorithms for real-world challenges.

Code :

Contains Duplicate:



The screenshot displays the LeetCode interface for the 'Contains Duplicate' problem. The submission is marked as 'Accepted' with 77/77 test cases passed. The runtime is 59 ms (beats 24.26%) and memory usage is 93.11 MB (beats 6.70%). The code is written in C++ and uses an unordered_set to check for duplicates.

```

1 class Solution {
2 public:
3     bool containsDuplicate(vector<int>& nums) {
4         unordered_set<int> s;
5         for(int n:nums)
6         {
7             s.insert(n);
8         }
9         if(nums.size()>s.size()) return true; else return false;
10    }
11 }
  
```

First Letter to Appear Twice:

Accepted 92 / 92 testcases passed
Ashwara Hasan submitted at Jan 12, 2026 14:29

Runtime
0 ms | Beat: 100.00%
8.27 MB | Beat: 59.93%

```

1 class Solution {
2 public:
3     char repeatedCharacter(string s) {
4         map<char, int> freq;
5         for (char ch : s) {
6             freq[ch]++;
7             if (freq[ch] > 1) return ch;
8         }
9     }
10 }

```

Testcase: Test Result

You must run your code first

Minimum Absolute Difference

Accepted 38 / 38 testcases passed
Ashwara Hasan submitted at Jan 12, 2026 14:49

Runtime
21 ms | Beat: 9.61%
36.66 MB | Beat: 63.84%

```

1 class Solution {
2 public:
3     vector<vector<int>> minimumAbsDifference(vector<int>& arr) {
4         sort(arr.begin(), arr.end());
5         vector<vector<int>> ans;
6         int target = arr[0];
7         for (int i = 1; i < arr.size(); i++) {
8             target = min(target, abs(arr[i] - arr[i-1]));
9         }
10        for (int i = 1; i < arr.size(); i++) {
11            if (abs(arr[i] - arr[i-1]) == target) {
12                ans.push_back({arr[i-1], arr[i]});
13            }
14        }
15        return ans;
16    }
17 }

```

Testcase: Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3 Case 4

Input
arr = [4, 2, 1, 3]

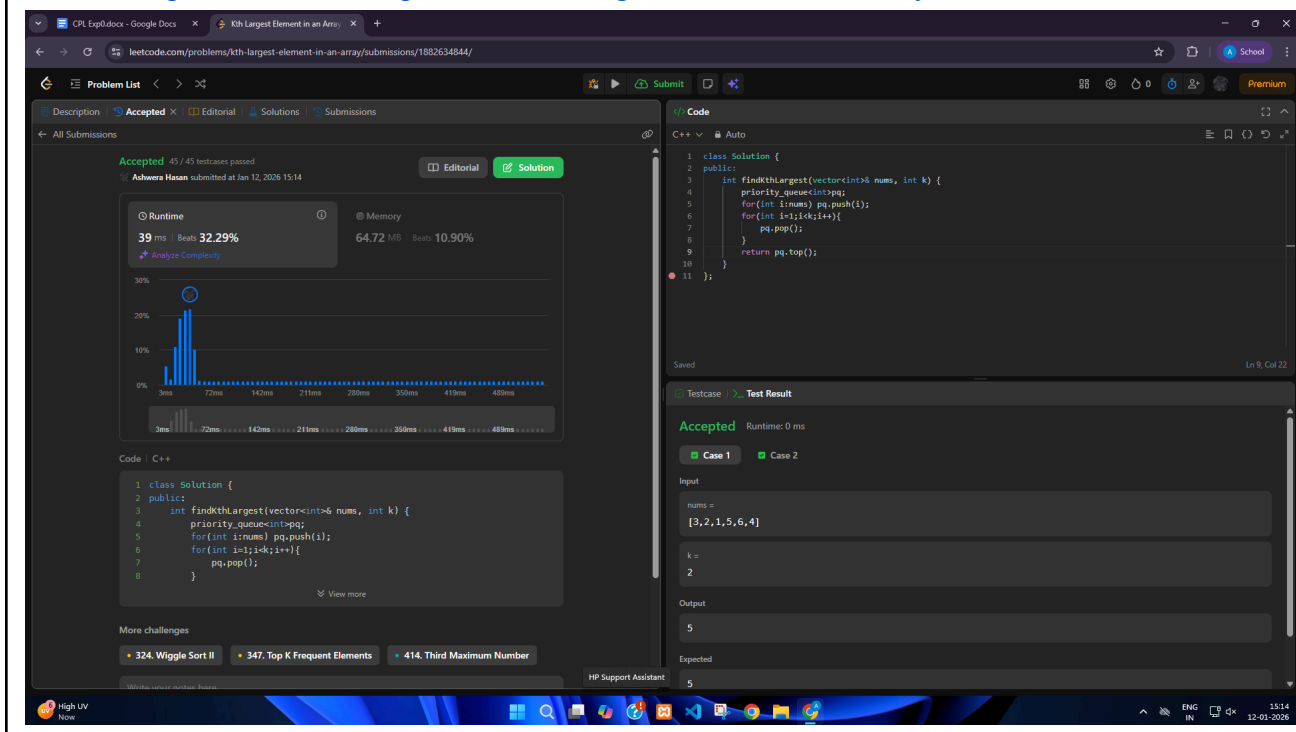
Output
[[1, 2], [2, 3], [3, 4]]

Expected
[[1, 2], [2, 3], [3, 4]]

Post Lab Subjective/Objective type Questions:

Create your account on Leetcode, solve and submit the following Leetcode problems:

LC 215: <https://leetcode.com/problems/kth-largest-element-in-an-array/>



The screenshot shows the LeetCode submission interface for problem 215, "Kth Largest Element in an Array". The submission is accepted, with a runtime of 39 ms and memory usage of 64.72 MB. The C++ code uses a priority queue to solve the problem. The test case shows an input array [3, 2, 1, 5, 6, 4] and k=2, resulting in an output of 5.

```

1 class Solution {
2 public:
3     int findKthLargest(vector<int>& nums, int k) {
4         priority_queue<int> pq;
5         for(int i:nums) pq.push(i);
6         for(int i=1; i<=k; i++){
7             pq.pop();
8         }
9         return pq.top();
10    }
11 };

```

Conclusion:

This experiment strengthened the foundations of data structures and algorithms and their implementation of the same in C++.

Signature of faculty in-charge with Date: