



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

WORKSHEET 1

Student Name: Hardik Jhamb

UID: 22BCS16504

Branch: BE-CSE

Section/Group: 22BCS_NTPP-602-A

Semester: 6th

Date of Performance:

Subject Name: AP LAB - II

Subject Code: 22CSP-351

1. **Aim:** Remove duplicates from a sorted array
Contains duplicate
Two Sum

2. Source Code:

```
class Solution {
public:
    bool isMirror(TreeNode* left, TreeNode* right) {
        if (!left && !right) return true;
        if (!left || !right) return false;
        return (left->val == right->val) && isMirror(left->left, right->right) && isMirror(left->right, right->left);
    }

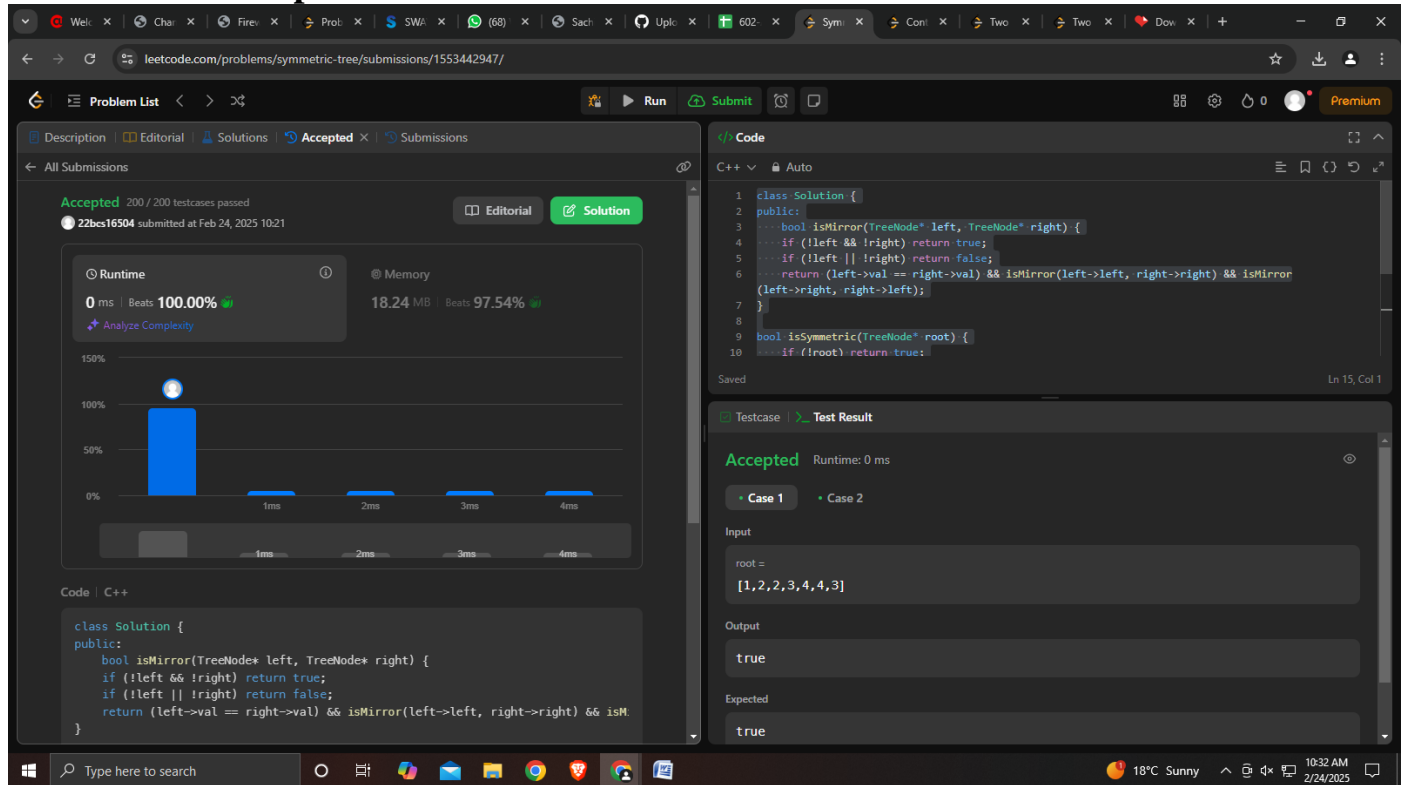
    bool isSymmetric(TreeNode* root) {
        if (!root) return true;
        return isMirror(root->left, root->right);
    }
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

3. Screenshots of outputs:



2. Aim: Given an integer array `nums`, return `true` if any value appears **at least twice** in the array, and return `false` if every element is distinct.

Source Code:

```
class Solution {
public:
    bool containsDuplicate(vector<int>& nums) {
        int n = nums.size();
        for (int i = 0; i < n - 1; i++) {
            for (int j = i + 1; j < n; j++) {
                if (nums[i] == nums[j])
                    return true;
            }
        }
        return false;
    }
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Screenshots of outputs:

217. Contains Duplicate Attempted

Easy Topics Companies

Given an integer array `nums`, return `true` if any value appears **at least twice** in the array, and return `false` if every element is distinct.

Example 1:
Input: `nums = [1,2,3,1]`
Output: `true`
Explanation:
The element 1 occurs at the indices 0 and 3.

Example 2:
Input: `nums = [1,2,3,4]`
Output: `false`
Explanation:
All elements are distinct.

Example 3:
Input: `nums = [1,1,1,3,3,4,3,2,4,2]`

```
1 public:
2     bool containsDuplicate(vector<int>& nums) {
3         int n = nums.size();
4         for (int i = 0; i < n - 1; i++) {
5             for (int j = i + 1; j < n; j++) {
6                 if (nums[i] == nums[j])
7                     return true;
8             }
9         }
10        return false;
11    }
12 }
```

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input
`nums = [1,2,3,1]`

Output
`true`

Expected
`true`



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

3. Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to `target`*.

You may assume that each input would have **exactly one solution**, and you may not use the *same* element twice.

You can return the answer in any order

Source Code:

```
class Solution{
public:
    vector<int> twoSum(vector<int>& nums, int target) {
        for (int i = 0; i < nums.size(); ++i) {
            for (int j = i + 1; j < nums.size(); ++j) {
                if (nums[i] + nums[j] == target) {
                    return {i, j};
                }
            }
        }
        return {};
    }
};
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

4. Screenshots of outputs:

The screenshot displays the LeetCode interface for the 'Two Sum' problem. The problem description on the left states: 'Given an array of integers `nums` and an integer `target`, return indices of the two numbers such that they add up to `target`. You may assume that each input would have **exactly one solution**, and you may not use the same element twice. You can return the answer in any order.' It includes three examples and constraints. The right panel shows a submission with a runtime of 41 ms (beats 33.12%) and memory of 14.10 MB (beats 76.83%). The 'Testcase' tab is active, showing Case 1 with `nums = [2, 7, 11, 15]` and `target = 9`.

1. Two Sum Solved

Easy Topics Companies Hint

Given an array of integers `nums` and an integer `target`, return indices of the two numbers such that they add up to `target`.

You may assume that each input would have **exactly one solution**, and you may not use the same element twice.

You can return the answer in any order.

Example 1:

Input: `nums = [2,7,11,15]`, `target = 9`
Output: `[0,1]`
Explanation: Because `nums[0] + nums[1] == 9`, we return `[0, 1]`.

Example 2:

Input: `nums = [3,2,4]`, `target = 6`
Output: `[1,2]`

Example 3:

Input: `nums = [3,3]`, `target = 6`
Output: `[0,1]`

Constraints:

60.4K 1.2K 2017 Online

Accepted 63 / 63 testcases passed
22bec16504 submitted at Jan 20, 2025 11:02

Runtime 41 ms | Beats 33.12%
Memory 14.10 MB | Beats 76.83%

Testcase Test Result

Case 1 Case 2 Case 3 +

`nums =`
`[2, 7, 11, 15]`

`target =`
`9`

Source