Description | Solution | Discuss (999+) | Submissions

1. Two Sum

Easy 32104 1019 Add to List Share

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
```

C++ | Autocomplete

```cpp
class Solution {
public:
    vector<int> twoSum(vector<int>& nums, int target) {
        vector<int> ans;

        for (int i = 0; i < nums.size(); i++)
        {
            for (int j = 1; j < nums.size(); j++)
            {
                if (nums[i] + nums[j] == target && i != j)
                {
                    ans.push_back(i);
                    ans.push_back(j);
                    return ans;
                }
            }
        }

        return ans;
    }
};
```

Testcase | Run Code Result | Debugger

Accepted Runtime: 0 ms

Your input
```
[2,7,11,15]
9
```

Output `[0,1]` Diff

Expected `[0,1]`