

<https://leetcode.com/problems/two-sum/description/>

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]`.

The screenshot displays the LeetCode interface for the "Two Sum" problem. The left sidebar shows the problem status as "Accepted" with a runtime of 43 ms, beating 44.65% of solutions, and a memory usage of 12.86 MB, beating 56.86%. A bar chart illustrates the distribution of runtimes. The main content area shows the C++ code for the solution, which uses a nested loop to find two numbers that sum to the target. The right sidebar shows the test results, indicating that the solution passed all test cases, including Case 3, with a runtime of 2 ms. The input for Case 3 is `nums = [3,3]` and `target = 6`, resulting in the output `[0,1]`.

Accepted

Antra_03 submitted at Jul 17, 2024 15:54

Runtime: 43 ms | Beats 44.65%
Memory: 12.86 MB | Beats 56.86%

Accepted

Runtime: 2 ms

Case 1 Case 2 Case 3

Input

nums = [3,3]

target = 6

Output

[0,1]

Expected

[0,1]

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