

<https://leetcode.com/problems/two-sum/submissions/1323936123/>

```
class Solution {
```

```
public:
```

```
    vector<int> twoSum(vector<int>& nums, int target) {
```

```
        vector<int> result={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){
```

```
                    result[0]=i;
```

```
                    result[1]=j;
```

```
                    return result;
```

```
                }
```

```
            }
```

```
        }
```

```
        return result;
```

```
    }
```

```
};
```

The screenshot displays the LeetCode interface for the 'Two Sum' problem. On the left, the problem description 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.'

Three examples are provided:

- 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: `2 <= nums.length <= 104`

The central panel shows the submission status as 'Accepted' with a runtime of 50 ms (Beats 32.55%) and memory usage of 12.96 MB (Beats 55.51%). A bar chart illustrates the runtime distribution across various time intervals.

The right panel shows the C++ code in the editor:

```
1 class Solution {
2 public:
3     vector<int> twoSum(vector<int>& nums, int target) {
4         vector<int> result={0,0};
5         for(int i=0;i<nums.size();i++){
6             for(int j=i+1;j<nums.size();j++){
7                 if(nums[i]+nums[j]==target){
8                     result[0]=i;
9                     result[1]=j;
10                    return result;
11                }
12            }
13        }
14        return result;
15    }
16};
```