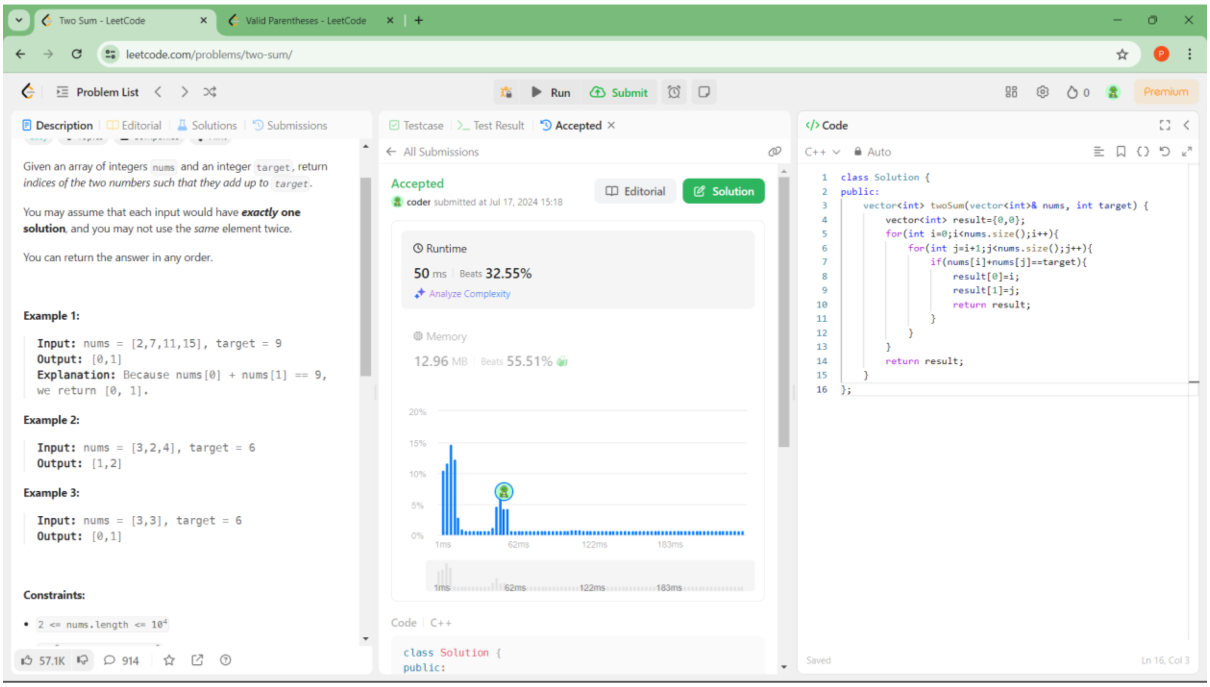
https://leetcode.com/problems/two-sum/submissions/1323936123/

```
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;
    }
};
```



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:

- `2 <= nums.length <= 10⁴`

Accepted

coder submitted at Jul 17, 2024 15:18

Runtime

50 ms | Beats 32.55%

Analyze Complexity

Memory

12.96 MB | Beats 55.51%

Code | C++

```
class Solution {
public:
```

C++ Auto

```
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;
    }
};
```