

一、题目说明

题目1. Two Sum, 计算2个数的和。难度是Easy! 但我懵了。

Given an array of integers, return **indices** of the two numbers such that they add up to a specific target.

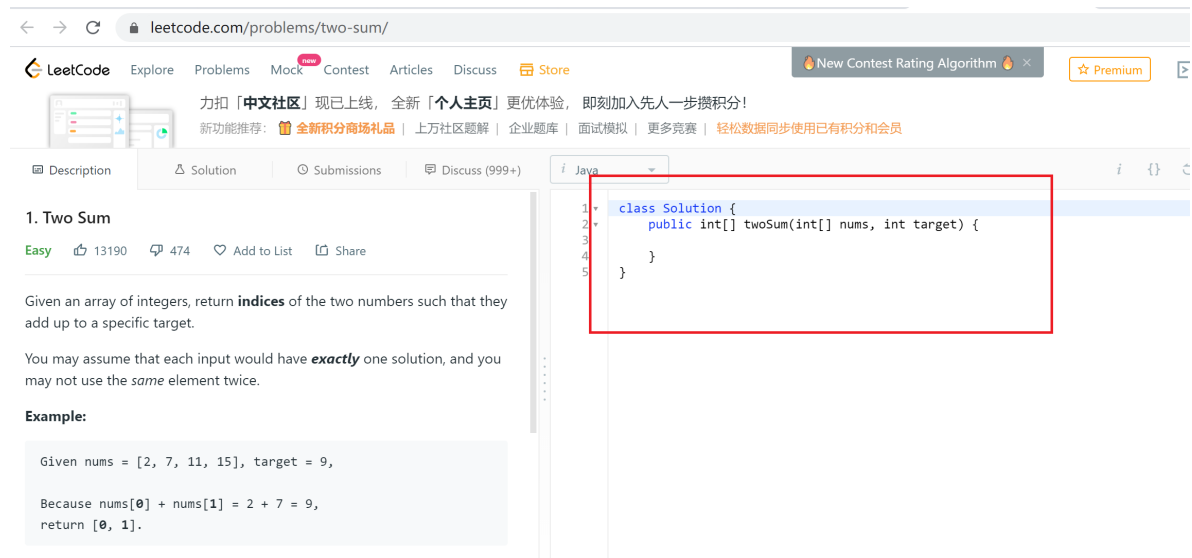
You may assume that each input would have **exactly** one solution, and you may not use the *same* element twice.

Example:

```
Given nums = [2, 7, 11, 15], target = 9,
```

```
Because nums[0] + nums[1] = 2 + 7 = 9,  
return [0, 1].
```

好在题目右边，有一个框，选择java的时候，会出现：

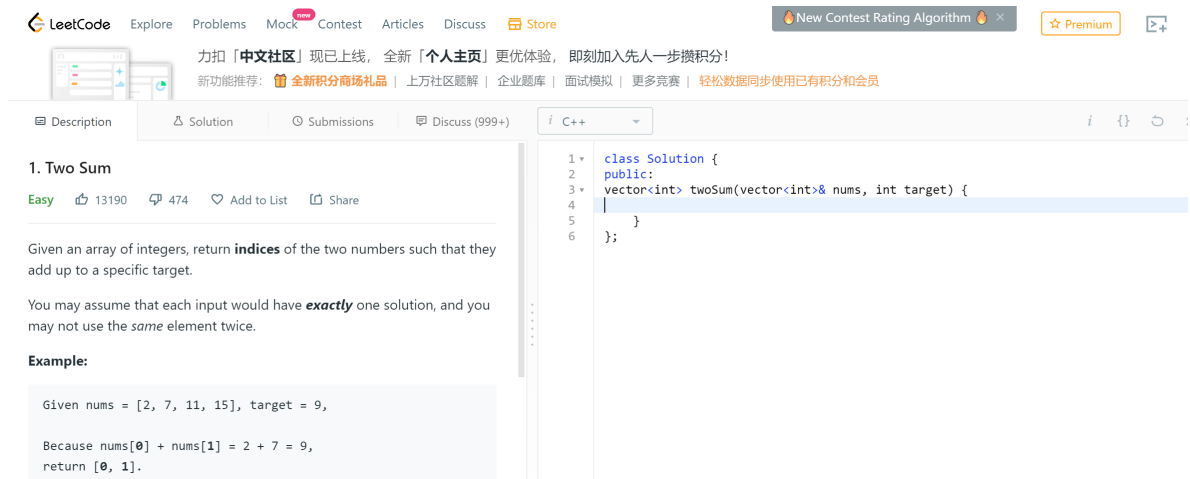


如果说用java写，我觉得还是写的出来的。

毕竟一直在用，而且对于里面的集合相关类都非常熟悉。

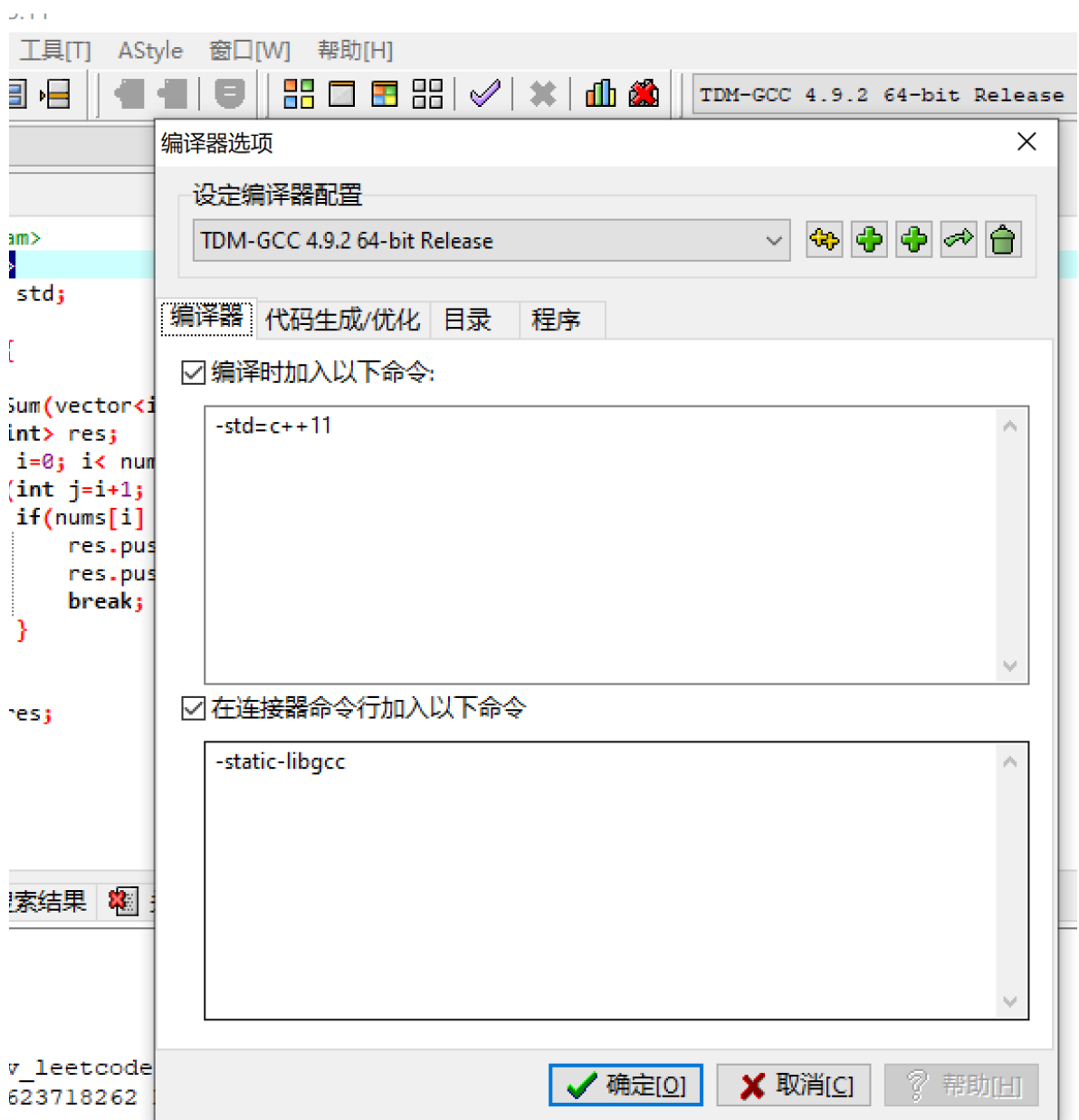
二、我的解答

但C++，我确实倍感吃力，主要是STL很少用，上来给了vector，我就已经无力了：



我在dev-c++写代码，遇到3个问题。

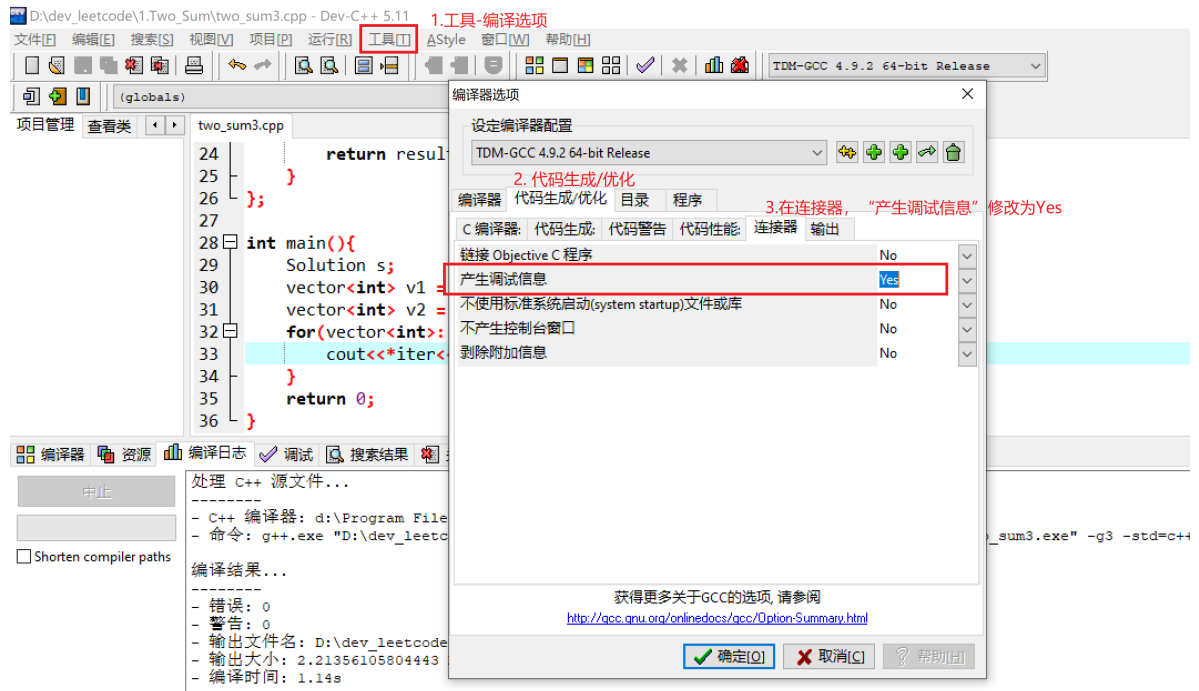
第1个是环境，需要配置 `-std=c++11`，见下图：



第2个，就是要

```
#include<vector>
#include<unordered_map>
```

第3个，需要增加调试：



看来要补的知识还很多。

三、疑问

在此，我有一个疑问，代码提交上去后，他们怎么测试？

是不是有test case：生成Solution对象，调用twoSum(),验证结果？

还需要学习啊！

四、总结

写了2个版本，第一个版本：

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

第二个版本，网上找的：

```
class Solution {
public:
    vector<int> twoSum(vector<int>& nums, int target) {
        unordered_map<int,int> mapping;
        vector<int> result;

        for(int i=0;i<nums.size();i++){
            mapping[nums[i]] = i;
        }

        for(int i=0; i< nums.size(); i++){
            const int gap = target - nums[i];

            if(mapping.find(gap) != mapping.end() && mapping[gap]>i){
                result.push_back(i);
                result.push_back(mapping[gap]);
                break;
            }
        }
        return result;
    }
};
```

上述代码，是需要提交的，如果在本地的dev-c++中，则需要增加main函数：

```
int main(){
    Solution s;
    vector<int> v = {2,7,11,15};
    vector<int> v2 = s.twoSum(v,9);
    for(vector<int>::iterator iter=v2.begin();iter!=v2.end();++iter)
        cout<<*iter<<" ";
    return 0;
}
```

内存差不多，但速度提升10倍：

LeetCode.com/problems/two-sum/submissions/

Two Sum III - Data structure design Subarray Sum Equals K

Two Sum IV - Input is a BST Two Sum Less Than K

Time Submitted	Status	Runtime	Memory	Language
a few seconds ago	Accepted	20 ms	10.4 MB	cpp
a day ago	Accepted	176 ms	9.3 MB	cpp

Your previous code was restored from your local storage. [Reset to default](#)

Testcase Run Code Result Debbuger **BETA**

Accepted Runtime: 4 ms

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

为了后面更快的刷题，本来我就非常熟悉c++，但不熟悉STL，two_sum我写了7遍。

差不多，可以做到“快”，“bug free”了。

为什么要写这么多遍??? 请参考我的另两篇博客：

[原来我也可以这么优秀（一）一个简单的极限测试](#)

[个人领会的成功秘诀-坚持中改进](#)