原题：

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**].

分析：

用一个map保存已经遍历过的数值和下标的映射关系。当遍历到一个新数值时，查找map中是否含有target需要的另一个数值，如果有则返回。时间复杂度O(n)。

class Solution {

public:

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

std::vector<int> result;

std::map<int, int> num\_map;

for (int i = 0; i < nums.size(); ++i) {

int other = target - nums[i];

if (num\_map.find(other) != num\_map.end()) {

result.push\_back(num\_map[other]);

result.push\_back(i);

}

num\_map[nums[i]] = i;

}

return result;

}

};