# 题目

给定一个整数数组和一个整数 k，你需要找到该数组中和为 k 的连续的子数组的个数。

**示例1:**

输入:nums = [1,1,1], k = 2

输出: 2 , [1,1] 与 [1,1] 为两种不同的情况。

**说明:**

数组的长度为 [1, 20,000]。

数组中元素的范围是 [-1000, 1000] ，且整数 k 的范围是 [-1e7, 1e7]。

# 分析

## 方法一：暴力破解

## 方法二：哈希表

class Solution {

public:

int subarraySum(vector<int>& nums, int k) {

unordered\_map<int,int> mp;

mp[0] = 1;

int count=0,sum=0;

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

{

//求和

sum += nums.at(i);

//查找是否存在等于sum的集合

if(mp.find(sum-k) != mp.end())

count += mp[sum-k];

mp[sum]++;

}

return count;

}

};

**复杂度：**

时间复杂度为O(n)；

空间复杂度为 O(n)。