

前缀和+哈希加速

题目描述：

560. 和为K的子数组

难度 中等

👍 1086



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：

1. 数组的长度为 [1, 20,000]。
2. 数组中元素的范围是 [-1000, 1000]，且整数 **k** 的范围是 [-1e7, 1e7]。

题目分析：

我们定义 $pre[i]$ 为 $[0..i]$ 里所有数的和，则 $pre[i]$ 可以由 $pre[i - 1]$ 递推而来，即：

$$pre[i] = pre[i - 1] + nums[i]$$

那么「 $[j..i]$ 这个子数组和为 k 」这个条件我们可以转化为

$$pre[i] - pre[j - 1] == k$$

简单移项可得符合条件的下标 j 需要满足

$$pre[j - 1] == pre[i] - k$$

所以我们考虑以 i 结尾的和为 k 的连续子数组个数时只要统计有多少个前缀和为 $pre[i] - k$ 的 $pre[j]$ 即可。我们建立哈希表 mp

C++简洁代码

```
class Solution {
public:
    int subarraySum(vector<int>& nums, int k) {
        unordered_map<int, int> mymap;
        mymap[0] = 1;
        int pre = 0, count = 0, n = nums.size();
        for (int i = 0; i < n; i++) {
            pre += nums[i];
            if (mymap.find(pre - k) != mymap.end()) {
                count += mymap[pre - k];
            }
            ++mymap[pre];
        }
        return count;
    }
};
```