560. Subarray Sum Equals K

Description | Hints | Submissions | Discuss | Solution

Pick One

Given an array of integers and an integer **k**, you need to find the total number of continuous subarrays whose sum equals to **k**.

Example 1:

```
Input:nums = [1,1,1], k = 2
Output: 2
```

Note:

1. The length of the array is in range [1, 20,000].
2. The range of numbers in the array is [-1000, 1000] and the range of the integer **k** is [-1e7, 1e7].

这个需要考虑负数的情况

```
public class L560 {
    public int subarraySum(int[] nums, int k) {
        /*
         * preSum存储的是到当前为止的总和，若前面有sum-k的和的位置，那么说明这两个数
         * 中间部分的和为k。result是到当前为止的连续和为k的个数。
         */
        int sum = 0 , result = 0;
        Map<Integer, Integer> preSum = new HashMap<Integer, Integer>();
        preSum.put(0, 1);
        for(int i = 0; i < nums.length; i ++) {
            sum += nums[i];
            if(preSum.containsKey(sum - k)) {
                result += preSum.get(sum - k);
            }
            //getOrDefault就是当preSum中存在sum这个key不，不存在则用0作为其value，存在即用其真实的value
            //这个的意思是从当前为止，前面数字连续之和等于sum的位置的个数，并不只有一个
            preSum.put(sum, preSum.getOrDefault(sum, 0) + 1);
        }
        return result;
    }
```