题目： 974. 和可被 K 整除的子数组

> 同余定理 （a - b) % p = k.....0 => a % p = b % p
> 负数 % 正数 正确表达： (a % b + b) % b = 正确余数

思想： 前缀和 + hash

```cpp
class Solution {
public:
    int subarraysDivByK(vector<int>& nums, int k) {
        //同余定理 如果 （a - b) % p = k.....0 => a % p = b % p
        unordered_map<int, int> hash = {{0, 1}}; // 哈希表：记录每个余数出现的次数
        int ret = 0;
        int x = 0; // 当前前缀和

        for (int i = 0; i < nums.size(); i++) {
            x += nums[i]; // 更新前缀和

            // 计算前缀和的余数，保证是非负数
            int mod = (x % k + k) % k;

            // 如果当前余数已存在，累加其出现次数
            if (hash.count(mod)) {
                ret += hash[mod];
            }

            // 更新当前余数的计数
            hash[mod]++;
        }

        return ret;

    }
};
```

题目： 525. 连续数组

```cpp
class Solution {
public:
    int findMaxLength(vector<int>& nums) {
        for (auto& e : nums) {
            if (e == 0)
                e = -1;
        }
        int maxlen = 0;
        int sum = 0;
        unordered_map<int, int> hash; // key: sum value : index
        hash[0] = -1;
        for (int i = 0; i < nums.size(); i++) {
            sum += nums[i];
            if (hash.count(sum))
                maxlen = max(maxlen,i - hash[sum]);
            else
                hash[sum] = i;
        }
        return maxlen;
    }
};
```