Atcode**D - Count Interval**[Editorial](https://atcoder.jp/contests/abc233/tasks/abc233_d/editorial) / 

[22周赛 - Virtual Judge (vjudge.net)](https://vjudge.net/contest/548413#problem/B)

我的思路：切入点：满足性质的情况，枚举。

//区间类枚举区间，然后求和，判断，统计数量，剪枝叶（如果一个步骤卡壳了5分钟就不要往这方面想了

当时想着如果是枚举区间的话应该也可以，然后超时，又开始想是不是要剪枝，想了很久还是想不到。是踩到出题人的陷阱了，之前没做过这个范围，以后看到要注意，1e5还可以

以后要注意2e5双循环会超时

题解思路：用前缀和表示公式aj – ai-1 = k,可以化成aj = k + ai-1;

答案精妙在，这个表达式的分式是两个变量和一个常量，于是可以用一个变量来表示另外一个变量，两种表示方式的值相同时，就是满足题意了，用map记录一种表示方式的数量，然后再枚举另外一种情况的值，这个值在map中对应的数量就是符合条件的情况的一部分，这样只要一重循环就行。

**D重做并错，有一个数组类型开错了，（写得太快？不集中？我觉得我已经够慢了）（debug思路**

// 求区间和：前缀和+区间枚举

// a[j] - a[i-1] == k;

// 数据大，变换 a[j] = a[i-1] + k;

// map维护a[i-1] + k;

// 是否存在等值a[j]

// 先map后a[j]

#include <iostream>

#include <cstring>

#include <cstdio>

#include <algorithm>

#include <map>

using namespace std;

typedef long long ll;

map<ll,ll>m;

const int N = 2e5 + 10;

ll f[N] ;

//

int main(){

ll n,k;

cin >> n>>k;

for(int i = 1;i <= n;i++){

scanf("%lld",&f[i]);

}

for(int i = 2;i <= n;i++){

f[i] += f[i-1];

}

ll cnt = 0;

for(int i = 1;i <= n;i++){

m[f[i-1] + k]++;

if(m.count(f[i])) cnt += m[f[i]];

}

cout << cnt << endl;

return 0;

}

二刷,秒了

// 1≤　N　≤2×105

// ∣≤109　∣Ai​∣　≤109

// ∣≤1015　∣K∣　≤1015

// All values in input are integers.

// 2e5的范围,只能一重循环

// All values in input are integers.

// In other words, how many pairs of integers (l,r) satisfy all of the conditions below?

// 1≤l≤r≤N

// ∑r​Ai​=K

// 求l到k的和,就会想到用前缀和,表示s[r] - s[i-1] == k

// 可以化成s[r] == k + s[i-1],然后使用可以使用map维护一边,枚举一边,来算结果

// l和r可以相等,先map后算结果

// 前缀和数组和原数组一起,s[N] ,map文件

// 检查范围

// 1≤　N　≤2×105

// ∣≤109　∣Ai​∣　≤109

// ∣≤1015　∣K∣　≤1015,

// All values in input are integers.

// k 1e15,变量要开long long

[云剪贴板 - 洛谷 | 计算机科学教育新生态 (luogu.com.cn)](https://www.luogu.com.cn/paste/7b4nme50)