OJ 更新 目前已經完成，如果有任何使用上的問題歡迎直接提出來(?)

23 . 老方與陣列

Submit (/problems/23/submissions/new)

Status (/problems/23/submissions)

Ranklist (/problems/23/ranklist)

Back to Problems List (/problems)

TopCoder

(/users/algo104)

顏少于-algo (/users/algo104)

> *可以都給我AC嗎:(*

User's AC Ratio

75.0% (6/8 (/problems/23/ranklist))

Submission's AC Ratio

45.0% (9 (/problems/23/submissions?filter_status=AC)/20 (/problems/23/submissions))

Tags

dc (/problems/tag/dc) dc10 (/problems/tag/dc10)

Description

老方剛學習 C++ 程式語言，最近他學到一種叫做陣列的資料結構，他有一個長度為 N 的陣列 $a_1, a_2, \ldots, a_N$，他想知道給定整數 x，有多少數對 (l, r) 滿足 $1 \leq l \leq r \leq n$ 使得區間 $[l, r]$ 中的元素和小於 x，也就是 $a_l + a_{l+1} + \cdots + a_r < x$，這讓老方遇到了困難，請你幫助他完成任務。

Input Format

輸入第一行包含兩個整數 $N, x (1 \leq N \leq 2 \times 10^5, |x| \leq 10^{14})$。
輸入第二行包含 N 個整數 $a_1, a_2, \ldots, a_N (|a_i| \leq 10^9)$。

Output Format

請輸出一個整數代表有多少區間的元素和小於 x。

Sample Input

copy

```
// Sample input 1
5 4
5 -1 3 4 -1

// Sample input 2
3 0
-1 2 -3

// Sample input 3
4 -1
-2 1 -2 3
```

Sample Output

copy

```
// Sample output 1
5

// Sample output 2
4

// Sample output 3
3
```

Hints

Problem Source

Codeforces

Solution (Click to toggle)

與區間和有關的題目常會使用前綴和 $(pref[i] = a_1 + \cdots + a_i)$ 來記錄資訊，如果想要取 $[i, j]$ 的區間和，就可以用 $pref[j] - pref[i-1]$ 來取得，這樣就可以用 $\mathcal{O}(N^2)$ 得到所有的區間和，但這裡還可以用分治將效率提升到 $\mathcal{O}(N \log N)$。

對於一個區間，把它分成兩段，那麼符合條件的區間會有幾種可能。

1. 全部在左邊那一段，或全部在右邊那一段： 遞迴下去就好了。
2. 區間的兩端分別在左邊那一段與右邊那一段上： 把兩邊的前綴和陣列由小到大排好。 枚舉右邊的終點，將左邊起點一路往右推。值得注意的是，因為兩邊的前綴和陣列已經排序過了，所以當右邊的終點向右推時，取到的區間和只會變大，左邊的起點就不需要再向左枚舉了。所以這個操作（先不管排序）可以在 $O(N)$ 時間內完成。

所以在一次遞迴裡要做： 1. 終止條件（個數 1） 2. 切陣列 3. 遞迴下去 4. 數第二種可能 5. 合併陣列（merge sort）

發現遞迴時間是 $T(N) = 2 \cdot T(N/2) + O(N) = O(N \log N)$，符合此題的限制條件。

Solution Code

```
#include<vector>
#include<iostream>
#include<utility>
#include<cstdint>

// data = {sorted array, # of inversions}
typedef std::pair<std::vector<int64_t>, int64_t> data;

// merge two data
data merge(data a, data b, const int64_t k){
    std::vector<int64_t> ret;
    auto &aa = a.first, &bb = b.first;
    int64_t ans = a.second + b.second;

    // calculating answer
    for(int i = 0, j = 0; j < bb.size(); j++){
        while(i < aa.size() && bb[j] - aa[i] >= k) i++;
        ans += aa.size()-i; // [i, aa.size()-1] is the good region
    }

    // merge-sorting
    for(int i = 0, j = 0; i < aa.size() || j < bb.size();)
        if(i < aa.size() && (j == bb.size() || aa[i] <= bb[j]))
            ret.push_back(aa[i++]);
        else ret.push_back(bb[j++]);

    return {ret, ans};
}

// recieves partial sum
data solve(std::vector<int64_t> a, const int64_t k){
    // base case
    if(a.size() <= 1) return {a, 0};

    // split a into two
    std::vector<int64_t> left, right;
    for(int i = 0; i < a.size()/2; i++)
        left.push_back(a[i]);
    for(int i = a.size()/2; i < a.size(); i++)
        right.push_back(a[i]);

    return merge(solve(left, k), solve(right, k), k);
}
```

```
int main() {
    // read input
    int n; int64_t k;
    std::cin >> n >> k;
    std::vector<int64_t> a(n+1);
    for(int i = 1; i <= n; i++) std::cin >> a[i], a[i] += a[i-1];

    std::cout << solve(a, k).second << std::endl;
}
```

Subtasks

No.	Testdata Range	Constraints	Score
1	0~2	範例測資	0
2	0~28	無特別限制	100

Testdata and Limits

No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	524288	65536	1 2
1	1000	524288	65536	1 2
2	1000	524288	65536	1 2
3	1000	524288	65536	2
4	1000	524288	65536	2
5	1000	524288	65536	2
6	1000	524288	65536	2
7	1000	524288	65536	2
8	1000	524288	65536	2
9	1000	524288	65536	2
10	1000	524288	65536	2
11	1000	524288	65536	2

No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
12	1000	524288	65536	2
13	1000	524288	65536	2
14	1000	524288	65536	2
15	1000	524288	65536	2
16	1000	524288	65536	2
17	1000	524288	65536	2
18	1000	524288	65536	2
19	1000	524288	65536	2
20	1000	524288	65536	2
21	1000	524288	65536	2
22	1000	524288	65536	2
23	1000	524288	65536	2
24	1000	524288	65536	2
25	1000	524288	65536	2
26	1000	524288	65536	2
27	1000	524288	65536	2
28	1000	524288	65536	2

Submit (/problems/23/submissions/new)

Status (/problems/23/submissions)

Ranklist (/problems/23/ranklist)

Back to Top