**小朋友排队**

问题描述

　　n 个小朋友站成一排。现在要把他们按身高从低到高的顺序排列，但是每次只能交换位置相邻的两个小朋友。  
  
　　每个小朋友都有一个不高兴的程度。开始的时候，所有小朋友的不高兴程度都是0。  
  
　　如果某个小朋友第一次被要求交换，则他的不高兴程度增加1，如果第二次要求他交换，则他的不高兴程度增加2（即不高兴程度为3），依次类推。当要求某个小朋友第k次交换时，他的不高兴程度增加k。  
  
　　请问，要让所有小朋友按从低到高排队，他们的不高兴程度之和最小是多少。  
  
　　如果有两个小朋友身高一样，则他们谁站在谁前面是没有关系的。

输入格式

　　输入的第一行包含一个整数n，表示小朋友的个数。  
　　第二行包含 n 个整数 H1 H2 … Hn，分别表示每个小朋友的身高。

输出格式

　　输出一行，包含一个整数，表示小朋友的不高兴程度和的最小值。

样例输入

3  
3 2 1

样例输出

9

样例说明

　　首先交换身高为3和2的小朋友，再交换身高为3和1的小朋友，再交换身高为2和1的小朋友，每个小朋友的不高兴程度都是3，总和为9。

数据规模和约定

　　对于10%的数据， 1<=n<=10；  
　　对于30%的数据， 1<=n<=1000；  
　　对于50%的数据， 1<=n<=10000；  
　　对于100%的数据，1<=n<=100000，0<=Hi<=1000000。

本题的C++参考代码如下：

#include<algorithm>

#include <iostream>

#include <cstdio>

#include <cstring>

#include<queue>

#include<set>

#include<stack>

#include<map>

#include<cmath>

#include<vector>

#include<string>

#define \_\_LL64 long long

#define clr(a) memset(a,0,sizeof a)

using namespace std;

const double pi = acos(-1.0);

const double eps = 1e-9;

const long long M = 1000000007;

const long long INF = 0x3f3f3f3f;

typedef long long LL;

typedef unsigned long long ULL;

const LL dir[4][2] = {0,1,-1,0,0,-1,1,0};

using namespace std;

LL \_buf[20];

inline long long read()

{

char c = getchar();

LL isMinus = 1;

long long x = 0;

while ( c!='-' && ( c<'0' || c>'9' )) c = getchar();

if (c=='-') {isMinus = -1;c= getchar();}

while ( c>='0' &&c<='9' ) {x = x\*10+c-'0';c=getchar();}

return x \* isMinus;

}

inline void write(long long num,LL flag = 1)

{

memset(\_buf,0,sizeof(\_buf));

if (num<0) {putchar('-');num=-num;}

LL p = 0;

if (num==0) p++;

else while (num){\_buf[p++]=num%10;num/=10;}

for (LL i=p-1;i>=0;--i) putchar(\_buf[i]+'0');

if (flag==1) puts("");

if (flag==0) printf(" ");

}

const LL Max = 111111;

int n,C[Max],a[Max],b[Max];

LL table[Max];

int lowbit(int x) {return x&(-x);}

void add(int x) { while(x<=n) { C[x]++;x+=lowbit(x); } }

int sum(int x)

{

int ans = 0;

while (x>0) {ans+=C[x];

x-=lowbit(x);}

return ans;

}

void hs()

{

sort(b,b+n);

for (int i=0;i<n;++i)

a[i] = lower\_bound(b,b+n,a[i]) - b + 1;

}

int main(){

for (int i=1;i<Max;++i)

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

while (scanf("%d",&n)!=EOF)

{

for (int i=0;i<n;++i) b[i] = a[i] = read();

hs();

clr (b);

clr (C);

for (int i=0;i<n;++i)

{

add(a[i]);

b[i] += i + 1- sum(a[i]);

}

clr(C);

for (int i=n-1;i>=0;i--)

{

add(a[i]);

b[i] += sum(a[i]-1);

}

LL ans = 0;

for (int i=0;i<n;++i)

ans += table[ b[i] ];

write(ans);

}

return 0;

}

本题的C参考代码如下：

#include<stdio.h>

int h[100100];

int un[100100];

int b[1000100];

int reb[1000100];

int Lowbit(int x){

return x&(x^(x-1));

}

int sum(int bit[], int idx){

int ret = 0;

while(idx > 0){

ret += bit[idx];

idx -= Lowbit(idx);

}

return ret;

}

void add(int bit[], int idx, int val){

while(idx < 1000100){

bit[idx] += val;

idx += Lowbit(idx);

}

}

long long uVal[100100];

int main(void){

int n, i;

scanf("%d", &n);

uVal[0] = 0;

for(i = 0; i < n; i++){

scanf("%d", &h[i]);

h[i]++;

uVal[i + 1] = uVal[i] + i + 1;

un[i] += i - sum(b, h[i]);

add(b, h[i], 1);

}

long long ans = 0;

for(i = n - 1; i >= 0; i--){

un[i] += sum(reb, h[i] - 1);

add(reb, h[i], 1);

ans += uVal[un[i]];

}

printf("%I64d\n", ans);

return 0;

}

本题的Java参考代码如下：

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main{

static int N = 100010;

static int MAX = 1000100;

static int[] C = new int[MAX];

static int[] S = new int[MAX];

static int[] b = new int[N];

static long[] total = new long[N];

static long ans;

static int[] num = new int[N];

static int T, s, t, i, j;

static int Lowbit(int x) {

return x & (-x);

}

static void add(int pos, int num, int[] P) {

while (pos <= MAX) {

P[pos] += num;

pos += Lowbit(pos);

}

}

static int Sum(int end, int[] P) {

int cnt = 0;

while (end > 0) {

cnt += P[end];

end -= Lowbit(end);

}

return cnt;

}

static void init() {

total[0] = 0;

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

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

}

}

public static void main(String[] args) throws IOException {

init();

BufferedReader buf = new BufferedReader(

new InputStreamReader(System.in));

T = Integer.parseInt(buf.readLine());

String[] str = buf.readLine().split(" ");

for (int j = 0; j < T; j++) {

num[j] = Integer.parseInt(str[j]);

add(num[j] + 1, 1, C);

b[j] = j - Sum(num[j], C);

b[j] -= Sum(num[j] + 1, C) - Sum(num[j], C) - 1;

}

ans = 0;

for (int j = T - 1; j > -1; --j) {

add(num[j] + 1, 1, S);

b[j] += Sum(num[j], S);

ans += total[b[j]];

}

System.out.println(ans);

}

}