

前缀和排序 + 双指针

人员

李佳声、窦浩轩、穆鹏宇、梁钰涵、胡赫轩、崔嘉睿、周子航、程晟泰、蔡云翔 到课

上周作业检查

Begin: 2024-11-23 08:30 CST

☆ 2024-1123 ~ 1124 三队上课

End: 2025-01-04 00:30 CST

Elapsed: 14:00:19:10

Running

Remaining: 27:15:40:49

OverviewProblemStatusRank (14:00:19:08)Discuss

SettingCloneUpdateDelete

Rank	Team	Score	Penalty	A 6 / 6	B 6 / 6	C 4 / 6	D 5 / 5
1	☆ hehe625 (刘佳赫)	4	7389	1:06:45:02	1:06:45:26	1:06:44:31	1:06:54:21
2	☆ yujiahaoa (Pswd com...)	4	40730	7:01:31:28	7:01:31:17	7:01:46:05	7:02:01:45
3	☆ Cst_AK_IOI (程晟泰)	4	55108	10:14:00:17	10:14:01:28	11:12:20:49	5:14:06:20
4	☆ ssine233 (?)	3	451	2:29:11	2:31:43		2:30:59
5	☆ syh123bc (_FL_)	2	188	1:22:30	1:46:09		
6	☆ huhexuan	2	36191			12:13:29:09 (-2)	12:13:02:29
7	☆ douhaoxuan	2	39019	13:13:03:15	13:13:16:15		

作业

https://vjudge.net/contest/678142

课堂表现

今天课上的第二题很多同学听思路都没有问题，但是做起来都比较吃力，回家要好好复习一下这个题目

课堂内容

CF448C Painting Fence

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 5000 + 5;
int w[maxn];

LL dfs(int l, int r) {
```

```

    if (l > r) return 0;
    if (l == r) return min(w[l], 1);

    int minn = 1e9+5;
    for (int i = l; i <= r; ++i) minn = min(minn, w[i]);

    int last = l-1; LL res = 0;
    for (int i = l; i <= r; ++i) {
        w[i] -= minn;
        if (!w[i]) res += dfs(last+1, i-1), last = i;
    }
    res += dfs(last+1, r);

    return min(res+minn, r-l+1LL);
}

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i];
    cout << dfs(1, n) << endl;
    return 0;
}

```

POJ 2566

```

#include <iostream>
#include <cmath>
#include <algorithm>

using namespace std;

const int maxn = 1000000 + 5;
struct node {
    int v, id;
} p[maxn];
bool cmp(node a, node b) { return a.v < b.v; }

int calc(int a, int b) { return abs(a - b); }

void solve(int n, int x) {
    int resl = -1, resr = -1, res = 2e9+100;
    for (int l = 1, r = 0; l <= n; ++l) {
        r = max(l-1, r);
        while (r+1<=n && p[r+1].v-p[l].v<=x) ++r;
        // l ~ r, l ~ r+1
        if (l<r && r<=n && calc(p[r].v-p[l].v,x)<calc(res,x)) {
            res = p[r].v-p[l].v, resl = p[l].id, resr = p[r].id;
        }
        if (l<r+1 && r+1<=n && calc(p[r+1].v-p[l].v,x)<calc(res,x)) {
            res = p[r+1].v-p[l].v, resl = p[l].id, resr = p[r+1].id;
        }
    }
}

```

```
    }  
}  
  
    cout << res << " " << min(resl, resr)+1 << " " << max(resl, resr) << endl;  
}  
  
int main()  
{  
    int n, k;  
    while (cin >> n >> k) {  
        if (!n && !k) break;  
        p[0].v = 0;  
        for (int i = 1; i <= n; ++i) {  
            int x; cin >> x; p[i] = {p[i-1].v+x, i};  
        }  
        ++n; p[n].v = 0, p[n].id = 0;  
        sort(p+1, p+n+1, cmp);  
  
        while (k -- ) { int x; cin >> x; solve(n, x); }  
    }  
    return 0;  
}
```