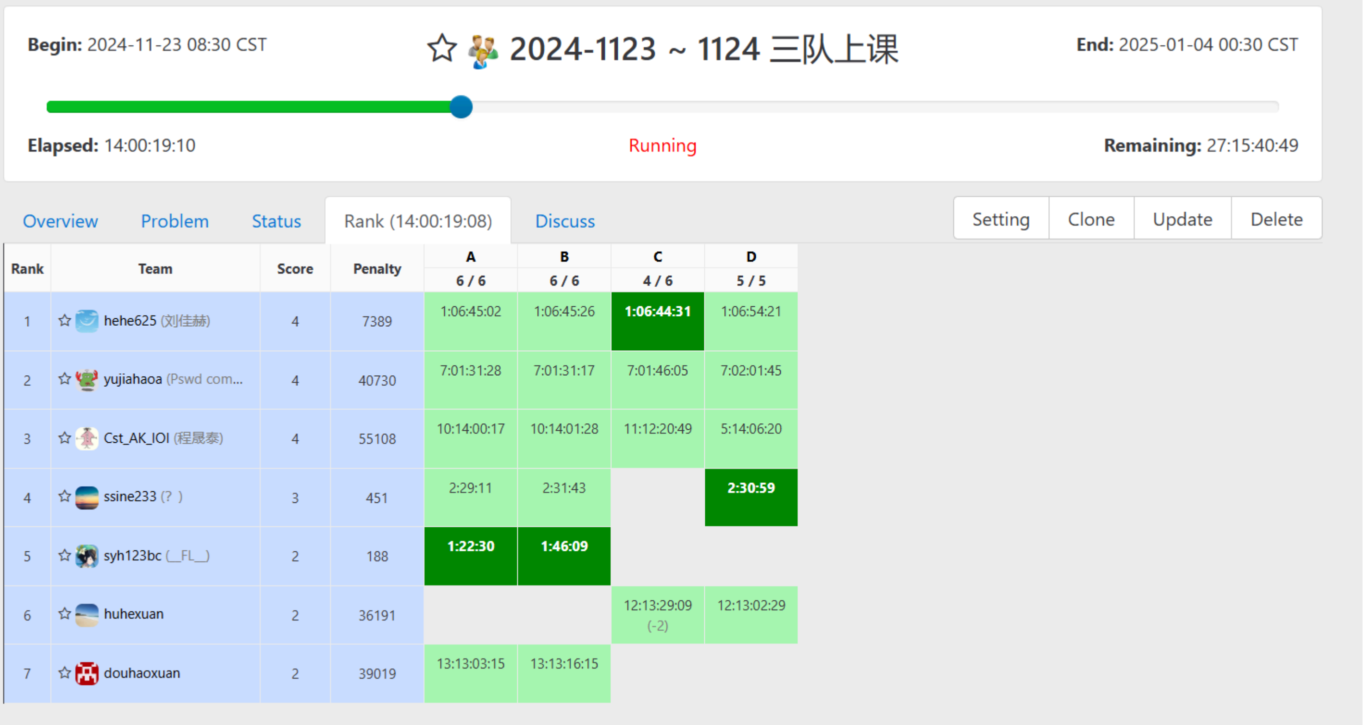
前缀和排序 + 二分

人员

左子毅、于珈浩 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

Rank	Team	Score	Penalty	A	B	C	D
				6 / 6	6 / 6	4 / 6	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

今天的 B 题是一个比较好的题，代码实现也比较复杂，同学们课下要好好复习一下 B 题。

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 <algorithm>
#include <cmath>

using namespace std;

const int maxn = 1e5 + 5;
struct node {
    int val, id;
    bool operator < (const node& p) const { return val < p.val; }
} p[maxn];

void solve(int n) {
    int t; cin >> t;
    int resl = -1, resr = -1, res = 2e9+10;
    for (int i = 0; i < n; i++) {
        int j = lower_bound(p+i+1, p+n+1, node{p[i].val+t, 0}) - p;
        // j: 右边第一个 >= p[i].val+t
        // i~j-1,   i~j
        if (i!=j-1 && abs(p[j-1].val - p[i].val - t) < abs(res - t)) {
            res = p[j-1].val - p[i].val, resl = p[i].id, resr = p[j-1].id;
        }
        if (j<=n && abs(p[j].val - p[i].val - t) < abs(res - t)) {
            res = p[j].val - p[i].val, resl = p[i].id, resr = p[j].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].val = p[0].id = 0;
        for (int i = 1; i <= n; ++i) {
            int x; cin >> x; p[i] = {p[i-1].val+x, i};
        }

        sort(p, p+n+1);

        while (k -- ) solve(n);
    }
    return 0;
}
```