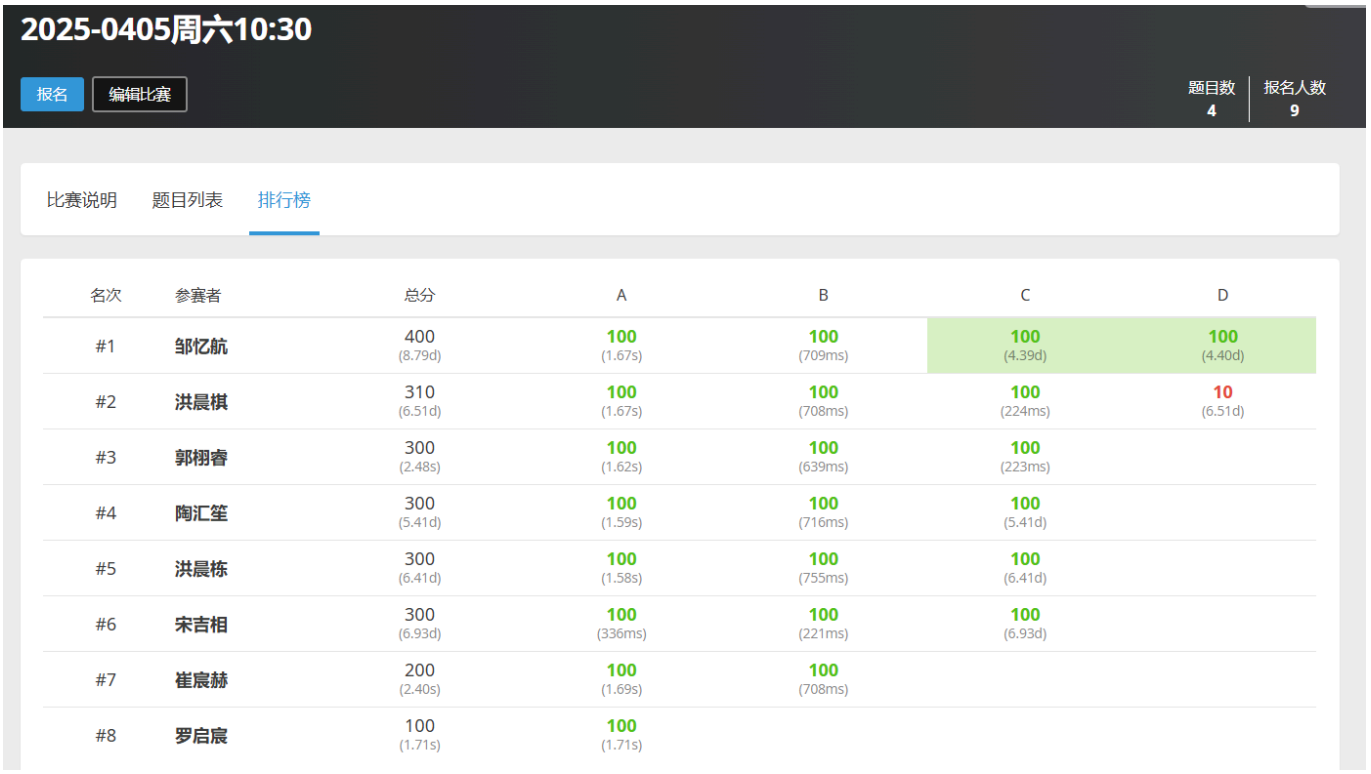
综合混练

人员

郭栩睿、邹忆航、宋吉相、李沛都、张曦月、陶汇笙 到课, 罗启宸、马敬杰、王恩泽 线上

作业检查

上周作业链接: https://www.luogu.com.cn/contest/240214



2025-0405周六10:30

报名 编辑比赛

题目数 4 | 报名人数 9

比赛说明 题目列表 排行榜

名次	参赛者	总分	A	B	C	D
#1	邹忆航	400 (8.79d)	100 (1.67s)	100 (709ms)	100 (4.39d)	100 (4.40d)
#2	洪晨棋	310 (6.51d)	100 (1.67s)	100 (708ms)	100 (224ms)	10 (6.51d)
#3	郭栩睿	300 (2.48s)	100 (1.62s)	100 (639ms)	100 (223ms)	
#4	陶汇笙	300 (5.41d)	100 (1.59s)	100 (716ms)	100 (5.41d)	
#5	洪晨栋	300 (6.41d)	100 (1.58s)	100 (755ms)	100 (6.41d)	
#6	宋吉相	300 (6.93d)	100 (336ms)	100 (221ms)	100 (6.93d)	
#7	崔宸赫	200 (2.40s)	100 (1.69s)	100 (708ms)		
#8	罗启宸	100 (1.71s)	100 (1.71s)			

作业

https://www.luogu.com.cn/contest/241311 (课上讲了 A ~ C 这些题, 课后作业是 D 题)

课堂表现

今天课上陶汇笙、郭栩睿 2 位同学做题表现比较好, 对这 2 位同学提出表扬, 其他同学要向这两位同学学习！！

课堂内容

P6180 [USACO15DEC] Breed Counting S

用 3 个一维数组, 或者用一个 3 行的 二维数组, 维护 1、2、3 每个数在每个位置是否出现

然后, 在维护前缀和数组, 就可以快速求一段区间内 1, 2, 3 的数量

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

```cpp
using namespace std;

const int maxn = 1e5 + 5;
int w[maxn], p[4][maxn];

int get_sum(int id, int l, int r) { return p[id][r] - p[id][l-1]; }

int main()
{
  int n, m; cin >> n >> m;
  for (int i = 1; i <= n; ++i) cin >> w[i];

  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= 3; ++j) p[j][i] = p[j][i-1];
    p[w[i]][i]++;
  }

  while (m -- ) {
    int l, r; cin >> l >> r;
    for (int i = 1; i <= 3; ++i) cout << get_sum(i, l, r) << " ";
    cout << endl;
  }
  return 0;
}
```

B4303 [蓝桥杯青少年组省赛 2024] 字母移位

考虑往左为正方向, 那么原问题本质上等价于:

- 第 1 个字符要变 a1-a2+a3-a4+a5-a6+...
- 第 2 个字符要变 -a2+a3-a4+a5-a6+...
- 第 3 个字符要变 a3-a4+a5-a6+...
- ...

所以, 可以维护一个数组, 数组的每一项分别为: a1, -a2, a3, -a4, a5, -a6, ...

第 i 个字符要进行的变化: 其实就是第 i 项一直到第 n 项的后缀和, 因此, 可以维护一个后缀和数组来 O(1) 求第 i 项字符的变化

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

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
char s[maxn];
int w[maxn], val[maxn];
LL suf[maxn];
```

```
char calc(char x, LL k) {
  k %= 26;
  if (k < 0) k += 26;

  x -= k;
  if (x < 'a') x += 26;
  return x;
}

int main()
{
  int n; cin >> n;
  cin >> (s+1);
  for (int i = 1; i <= n; ++i) {
    cin >> w[i];
    if (i&1) val[i] = w[i];
    else val[i] = -w[i];
  }

  for (int i = n; i >= 1; --i) suf[i] = suf[i+1] + val[i];

  for (int i = 1; i <= n; ++i) cout << calc(s[i], suf[i]);
  cout << endl;
  return 0;
}
```

P4086 [USACO17DEC] My Cow Ate My Homework S

题目本意是让求: 2~n, 3~n, 4~n, ..., n-1~n 区间中, 在每个区间都去掉一个最低分的情况下, 哪种情况下的区间平均值最大

因此, 可以 O(n) 维护一个 suf[i] 的后缀和数组 和一个 suf_min[i] 的后缀最小值数组

- suf[i] 代表: 区间 i~n 的区间和
- suf_min[i] 代表: 区间 i~n 的最小值

那么区间 i~n 在去掉一个最低分时区间的平均值是: (suf[i] - suf_min[i]) / (n-i) -> 可以 O(1) 求

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

using namespace std;

const int maxn = 1e5 + 5;
int w[maxn];
int suf_sum[maxn], suf_min[maxn];

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) cin >> w[i];
```

```
  suf_min[n+1] = 10000 + 5;
  for (int i = n; i >= 1; --i) {
    suf_sum[i] = suf_sum[i+1] + w[i];
    suf_min[i] = min(suf_min[i+1], w[i]);
  }

  double maxx_avg = -1.0;
  for (int k = 1; k <= n-2; ++k) {
    double t = 1.0*(suf_sum[k+1]-suf_min[k+1]) / (n-k-1);
    maxx_avg = max(maxx_avg, t);
  }

  for (int k = 1; k <= n-2; ++k) {
    double t = 1.0*(suf_sum[k+1]-suf_min[k+1]) / (n-k-1);
    if (t == maxx_avg) cout << k << endl;
  }
  return 0;
}
```