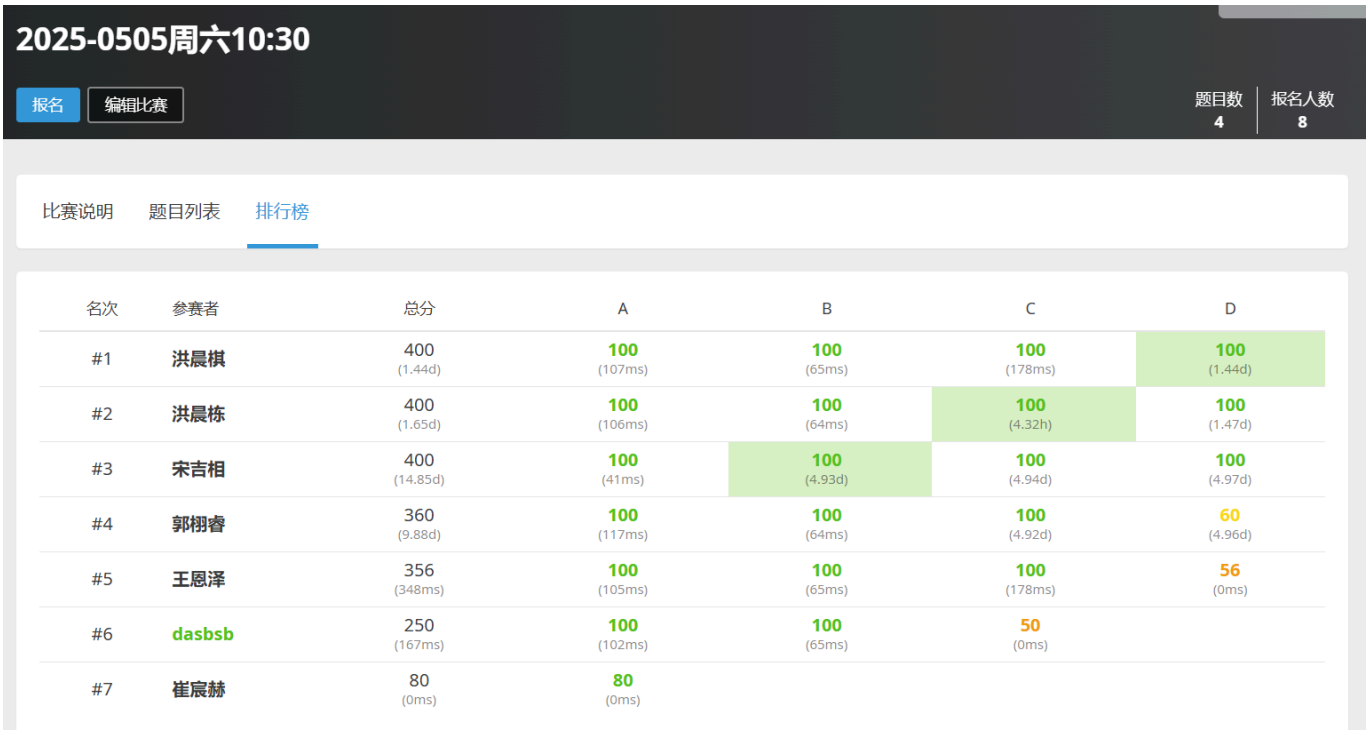
二分查找

人员

洪晨栋、洪晨棋、郭栩睿、宋吉相、陶汇笙、崔宸赫、王恩泽、罗启宸、邹忆航 到课, 马敬杰 线上

作业检查

上周作业链接: https://www.luogu.com.cn/contest/245142



2025-0505周六10:30

报名 编辑比赛

题目数 4 报名人数 8

比赛说明 题目列表 排行榜

名次	参赛者	总分	A	B	C	D
#1	洪晨棋	400 (1.44d)	100 (107ms)	100 (65ms)	100 (178ms)	100 (1.44d)
#2	洪晨栋	400 (1.65d)	100 (106ms)	100 (64ms)	100 (4.32h)	100 (1.47d)
#3	宋吉相	400 (14.85d)	100 (41ms)	100 (4.93d)	100 (4.94d)	100 (4.97d)
#4	郭栩睿	360 (9.88d)	100 (117ms)	100 (64ms)	100 (4.92d)	60 (4.96d)
#5	王恩泽	356 (348ms)	100 (105ms)	100 (65ms)	100 (178ms)	56 (0ms)
#6	dasbsb	250 (167ms)	100 (102ms)	100 (65ms)	50 (0ms)	
#7	崔宸赫	80 (0ms)	80 (0ms)			

作业

https://www.luogu.com.cn/contest/245869 (课上讲了 A ~ B 这些题, 课后作业是 C D 题)

课堂表现

今天课上讲的二分查找并不难, 但是lower_bound和upper_bound, 如果同学们不能做到熟练掌握的话, 很容易用错。

同学们课下要多花一点时间, 搞清楚 2 个的区别。

课堂内容

P3131 [USACO16JAN] Subsequences Summing to Sevens S

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

using namespace std;
```

```
typedef long long LL;
const int maxn = 5e4 + 5;
int w[maxn];
LL pre[maxn];
int p[10], s[10];

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) cin >> w[i];
  for (int i = 1; i <= n; ++i) {
    pre[i] = pre[i-1] + w[i]; pre[i] %= 7;
  }

  for (int i = 0; i <= 6; ++i) p[i] = s[i] = -1;
  for (int i = 0; i <= n; ++i) {
    int t = pre[i];
    if (p[t] == -1) p[t] = i;
  }
  for (int i = n; i >= 0; --i) {
    int t = pre[i];
    if (s[t] == -1) s[t] = i;
  }

  int res = 0;
  for (int i = 0; i <= 6; ++i) {
    int l = p[i], r = s[i];
    if (l==-1 || r==-1) continue;
    res = max(res, r-l);
  }
  cout << res << endl;
  return 0;
}
```

P2249 【深基13.例1】查找

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

using namespace std;

const int maxn = 1e6 + 5;
int w[maxn];

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

    while (t -- ) {
        int x; cin >> x;
        int pos = lower_bound(w+1, w+n+1, x) - w;
```

```
        cout << (w[pos]==x ? pos : -1) << " ";
    }

    return 0;
}
```

T476464 简简单单的二分查找模板检测

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

using namespace std;

const int maxn = 666666 + 5;
int w[maxn];

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

    int t; cin >> t;
    while (t -- ) {
        int x; cin >> x;
        int low = lower_bound(w+1, w+n+1, x) - w;
        int up = upper_bound(w+1, w+n+1, x) - w;
        // 1 ~ n+1
        if (low == 1) cout << -1 << " ";
        else cout << low-1 << " ";

        if (w[low] == x) cout << low << " ";
        else cout << -1 << " ";

        if (w[up-1] == x) cout << up-1 << " ";
        else cout << -1 << " ";

        if (up == n+1) cout << -1 << " ";
        else cout << up << " ";

        cout << endl;
    }
}

int main()
{
    int T; cin >> T;
    while (T -- ) solve();
    return 0;
}
```