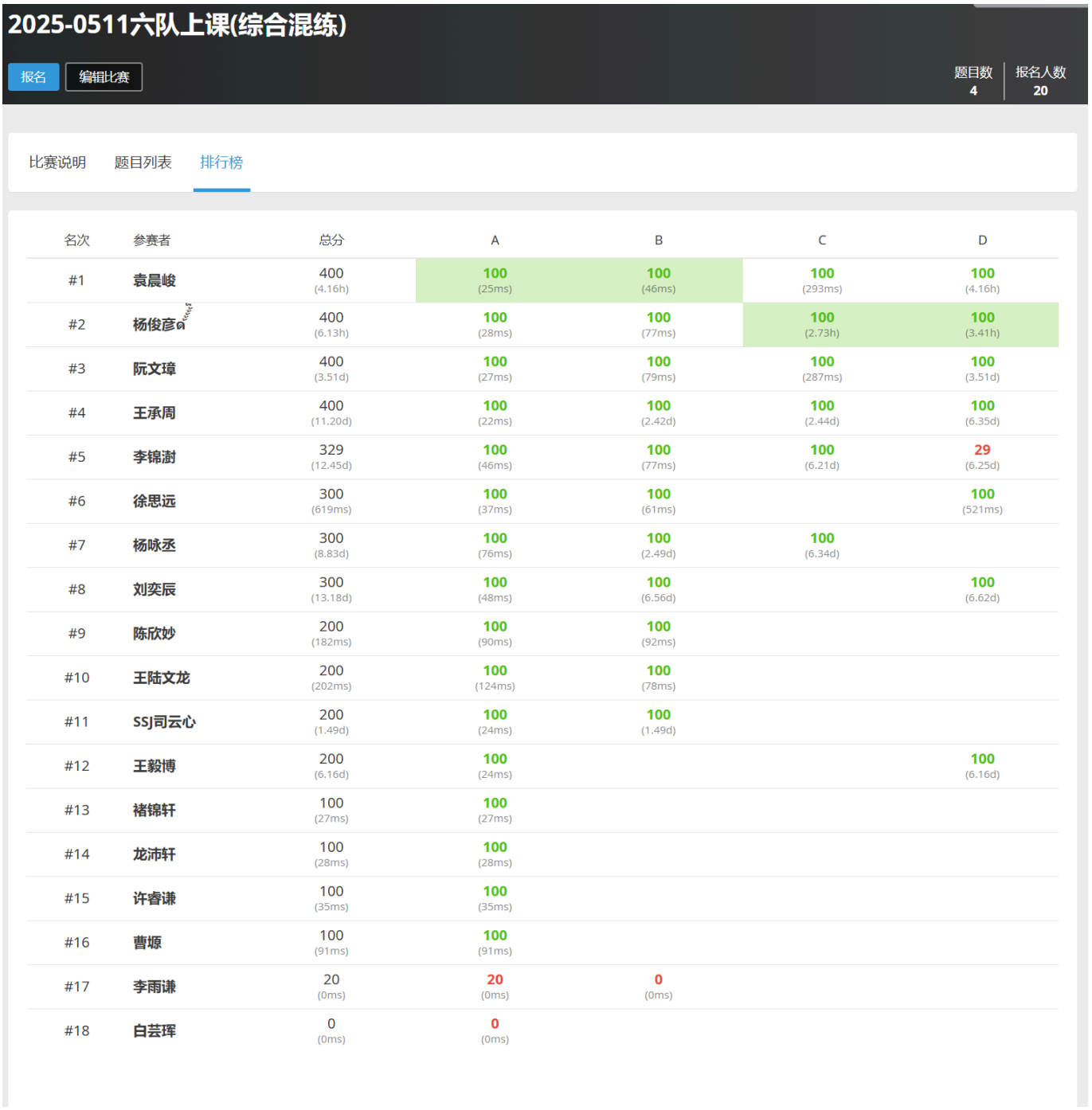
综合混练

人员

徐思远、刘奕辰、白芸珲、王博涵、袁晨峻、周治润 到课, 龙沛轩、杨俊彦、杨咏丞、王陆文龙 线上

上周作业检查

https://www.luogu.com.cn/contest/246063



2025-0511六队上课(综合混练)

报名 编辑比赛 题目数 4 报名人数 20

比赛说明 题目列表 排行榜

名次	参赛者	总分	A	B	C	D
#1	袁晨峻	400 (4.16h)	100 (25ms)	100 (46ms)	100 (293ms)	100 (4.16h)
#2	杨俊彦	400 (6.13h)	100 (28ms)	100 (77ms)	100 (2.73h)	100 (3.41h)
#3	阮文璋	400 (3.51d)	100 (27ms)	100 (79ms)	100 (287ms)	100 (3.51d)
#4	王承周	400 (11.20d)	100 (22ms)	100 (2.42d)	100 (2.44d)	100 (6.35d)
#5	李锦澍	329 (12.45d)	100 (46ms)	100 (77ms)	100 (6.21d)	29 (6.25d)
#6	徐思远	300 (619ms)	100 (37ms)	100 (61ms)		100 (521ms)
#7	杨咏丞	300 (8.83d)	100 (76ms)	100 (2.49d)	100 (6.34d)	
#8	刘奕辰	300 (13.18d)	100 (48ms)	100 (6.56d)		100 (6.62d)
#9	陈欣妙	200 (182ms)	100 (90ms)	100 (92ms)		
#10	王陆文龙	200 (202ms)	100 (124ms)	100 (78ms)		
#11	SSJ司云心	200 (1.49d)	100 (24ms)	100 (1.49d)		
#12	王毅博	200 (6.16d)	100 (24ms)			100 (6.16d)
#13	褚锦轩	100 (27ms)	100 (27ms)			
#14	龙沛轩	100 (28ms)	100 (28ms)			
#15	许睿谦	100 (35ms)	100 (35ms)			
#16	曹塬	100 (91ms)	100 (91ms)			
#17	李雨谦	20 (0ms)	20 (0ms)	0 (0ms)		
#18	白芸珲	0 (0ms)	0 (0ms)			

作业

https://www.luogu.com.cn/contest/247001 (课上讲了 A ~ C 题, 课后作业是 D E 题)

课堂表现

今天的 A 题和 B 题相对复杂一些, 同学们课上普遍做的不是特别好, 课下要多想一想, 好好写一写这两道题

课堂内容

P11272 「Diligent-OI R1 B」DlgtArray

1. 当 k > r-l 时, 无解
2. 当 k == r-l 时, [l,r] 这个区间里可以是 全1 或者是 r-l个1
3. 当 k < r-l 时, [l,r] 这个区间里必须是 k个1

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

using namespace std;

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

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

int main()
{
  int n, T; cin >> n >> T;
  for (int i = 1; i <= n; ++i) scanf("%d",&w[i]), p[i] = p[i-1] + w[i];
  while (T -- ) {
    int l, r, k; scanf("%d%d%d",&l,&r,&k);
    if (k > r-l) cout << -1 << "\n";
    else {
      if (k==r-l && get_sum(l,r)==r-l+1) cout << 0 << "\n";
      else cout << abs(k-get_sum(l,r)) << "\n";
    }
  }
  return 0;
}
```

U537160 move

bfs, 把每个字母对应的位置存下来, 一个字母只会做一次传送操作

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

using namespace std;

const int maxn = 2000 + 5;
char s[maxn][maxn];
int dis[maxn][maxn];
```

```cpp
int get_value(char x) { return x-'a'+1; }
bool f[30];
struct node {
  int x, y;
};
vector<node> vec[30];
int n, m, sx, sy, ex, ey;
int dx[] = {-1, 1, 0, 0}, dy[] = {0, 0, -1, 1};

void bfs() {
  memset(dis, -1, sizeof(dis));
  queue<node> q; q.push({sx,sy}); dis[sx][sy] = 0;

  while (!q.empty()) {
    node u = q.front(); q.pop();
    int x = u.x, y = u.y;
    for (int i = 0; i < 4; ++i) {
      int nx = x+dx[i], ny = y+dy[i];
      if (nx>=1 && nx<=n && ny>=1 && ny<=m && s[nx][ny]!='#' && dis[nx][ny]==-1) {
        q.push({nx,ny}); dis[nx][ny] = dis[x][y]+1;
      }
    }

    if (islower(s[x][y]) && !f[get_value(s[x][y])]) {
      int t = get_value(s[x][y]); f[t] = true;
      for (node it : vec[t]) {
        int nx = it.x, ny = it.y;
        if (dis[nx][ny] == -1) q.push({nx,ny}), dis[nx][ny] = dis[x][y]+1;
      }
    }
  }
}

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

  for (int i = 1; i <= n; ++i) {
    for (int j = 1; j <= m; ++j) {
      if (s[i][j] == 'S') sx = i, sy = j;
      if (s[i][j] == 'G') ex = i, ey = j;
      if (islower(s[i][j])) vec[get_value(s[i][j])].push_back({i, j});
    }
  }

  bfs();

  cout << dis[ex][ey] << endl;
  return 0;
}
```

P8755 [蓝桥杯 2021 省 AB2] 负载均衡

维护 n 个优先队列, 每个优先队列内维护的是该计算机处理的任务, 按照结束时间排序

每次处理一个 (a,b,c,d) 的任务时, 可以先把 b 这个优先队列中, 结束时间 <=a 的任务处理完, 然后再判断能否处理当前任务

如果能处理当前任务, 就加入到 b 这个优先队列中

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

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
int w[maxn];
struct node {
  int tm, value;
  friend bool operator < (node p, node q) { return p.tm < q.tm; }
  friend bool operator > (node p, node q) { return p.tm > q.tm; }
};
priority_queue<node, vector<node>, greater<node>> q[maxn];

int main()
{
  int n, m; cin >> n >> m;
  for (int i = 1; i <= n; ++i) cin >> w[i];
  while (m -- ) {
    int a, b, c, d; cin >> a >> b >> c >> d;
    while (!q[b].empty() && q[b].top().tm<=a) {
      w[b] += q[b].top().value; q[b].pop();
    }

    if (w[b] >= d) {
      w[b] -= d; q[b].push({a+c, d});
      cout << w[b] << endl;
    } else cout << -1 << endl;
  }
  return 0;
}
```

B3833 [NICA #2] 爱与不爱

先把每个数变成 2 的幂次方, 例如: 把 11->8, 把 88->64 等

然后用 multiset 维护, 每次把 最大的 和 最小的 选出来修改即可

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

using namespace std;
```

```cpp
typedef long long LL;

int main()
{
  vector<int> vec;
  for (int i = 1; i <= 1000000000; i *= 2) vec.push_back(i);

  multiset<int> s;
  int n; cin >> n;
  while (n -- ) {
    int x; cin >> x;
    auto it = upper_bound(vec.begin(), vec.end(), x);
    --it;
    s.insert(*it);
  }

   while (true) {
    int l = *s.begin(), r = *s.rbegin();
    if (l < r/2) {
      s.erase(s.find(l)), s.erase(s.find(r));
      s.insert(2*l), s.insert(r/2);
    } else break;
  }

  LL res = 0;
  for (int i : s) res += i;
  cout << res << endl;
  return 0;
}
```