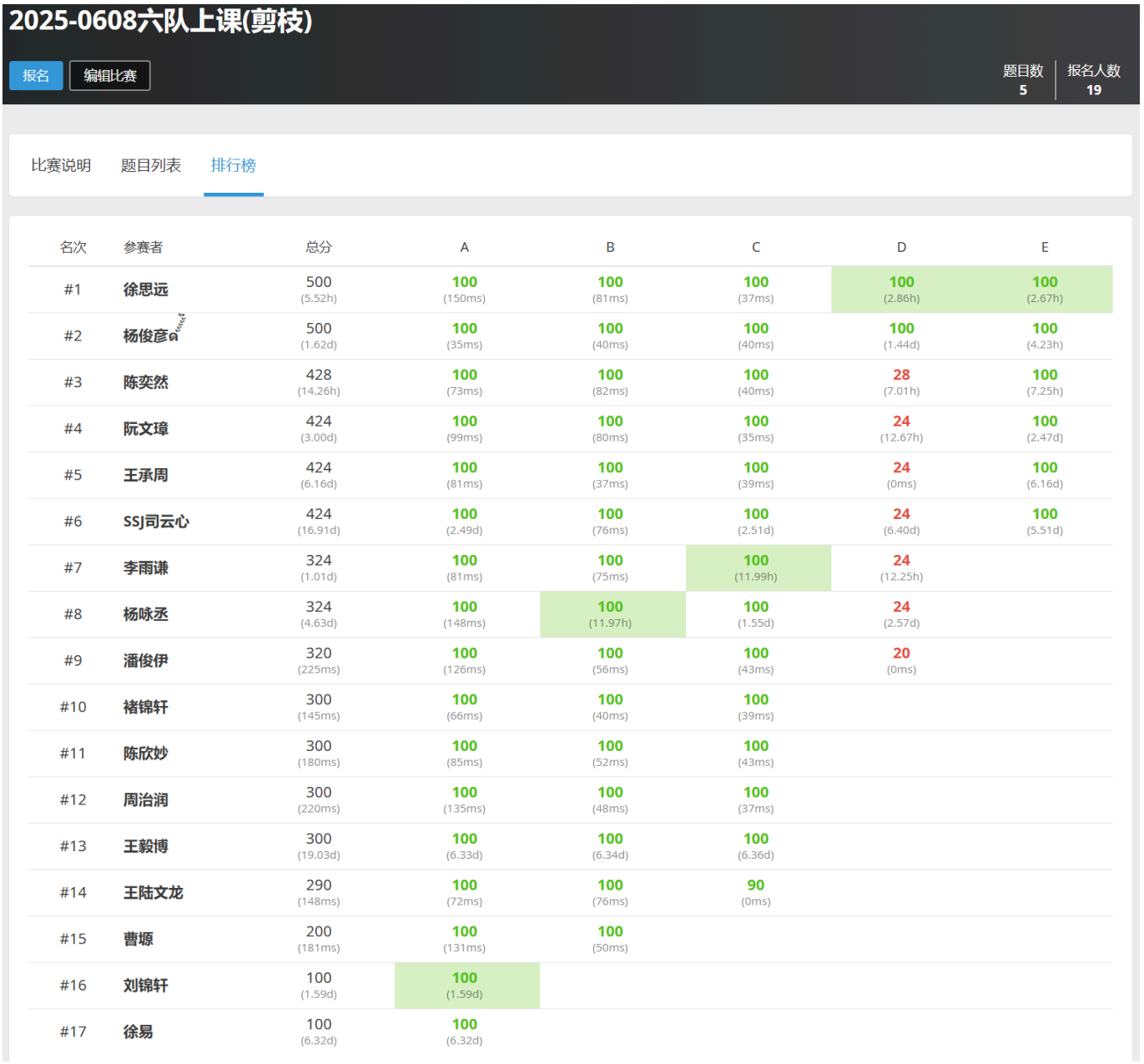
并查集

人员

褚锦轩、许睿谦、王承周、司云心、褚锦轩 到课, 阮文璋 线上

上周作业检查

https://www.luogu.com.cn/contest/251032



2025-0608六队上课(剪枝)

报名 编辑比赛

题目数 5 | 报名人数 19

比赛说明 题目列表 排行榜

名次	参赛者	总分	A	B	C	D	E
#1	徐思远	500 (5.52h)	100 (150ms)	100 (81ms)	100 (37ms)	100 (2.86h)	100 (2.67h)
#2	杨俊彦	500 (1.62d)	100 (35ms)	100 (40ms)	100 (40ms)	100 (1.44d)	100 (4.23h)
#3	陈奕然	428 (14.26h)	100 (73ms)	100 (82ms)	100 (40ms)	28 (7.01h)	100 (7.25h)
#4	阮文璋	424 (3.00d)	100 (99ms)	100 (80ms)	100 (35ms)	24 (12.67h)	100 (2.47d)
#5	王承周	424 (6.16d)	100 (81ms)	100 (37ms)	100 (39ms)	24 (0ms)	100 (6.16d)
#6	SSJ司云心	424 (16.91d)	100 (2.49d)	100 (76ms)	100 (2.51d)	24 (6.40d)	100 (5.51d)
#7	李雨谦	324 (1.01d)	100 (81ms)	100 (75ms)	100 (11.99h)	24 (12.25h)	
#8	杨咏丞	324 (4.63d)	100 (148ms)	100 (11.97h)	100 (1.55d)	24 (2.57d)	
#9	潘俊伊	320 (225ms)	100 (126ms)	100 (56ms)	100 (43ms)	20 (0ms)	
#10	褚锦轩	300 (145ms)	100 (66ms)	100 (40ms)	100 (39ms)		
#11	陈欣妙	300 (180ms)	100 (85ms)	100 (52ms)	100 (43ms)		
#12	周治润	300 (220ms)	100 (135ms)	100 (48ms)	100 (37ms)		
#13	王毅博	300 (19.03d)	100 (6.33d)	100 (6.34d)	100 (6.36d)		
#14	王陆文龙	290 (148ms)	100 (72ms)	100 (76ms)	90 (0ms)		
#15	曹塬	200 (181ms)	100 (131ms)	100 (50ms)			
#16	刘锦轩	100 (1.59d)	100 (1.59d)				
#17	徐易	100 (6.32d)	100 (6.32d)				

作业

https://www.luogu.com.cn/contest/252013 (课上讲了 A ~ D 题, 课后选做作业是 E 题, 必做作业是 F 题)

课堂表现

今天课上讲了并查集的内容, 并查集的代码非常简短, 但是思想非常重要, 同学们课下要好好再复习一下 A B C 三道题。

课堂内容

P2040 打开所有的灯

每个灯如果重复两次的话, 等于没操作

所有每个灯只有 动一次 或者 没动 两种情况, 因此可以 2^n 枚举所有可能即可。

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

using namespace std;

const int maxn = 5 + 5;
int w[maxn][maxn], a[maxn][maxn];
bool st[maxn][maxn];
int dx[] = {-1, 1, 0, 0}, dy[] = {0, 0, -1, 1};

bool check() {
  for (int i = 0; i < 3; ++i) {
    for (int j = 0; j < 3; ++j) a[i][j] = w[i][j];
  }

  for (int i = 0; i < 3; ++i) {
    for (int j = 0; j < 3; ++j) {
      if (st[i][j]) {
        a[i][j] = 1 - a[i][j];
        for (int k = 0; k < 4; ++k) {
          int ni = i+dx[k], nj = j+dy[k];
          if (ni>=0 && ni<3 && nj>=0 && nj<3) a[ni][nj] = 1 - a[ni][nj];
        }
      }
    }
  }

  for (int i = 0; i < 3; ++i) {
    for (int j = 0; j < 3; ++j) {
      if (!a[i][j]) return false;
    }
  }
  return true;
}

int main()
{
  for (int i = 0; i < 3; ++i) {
    for (int j = 0; j < 3; ++j) cin >> w[i][j];
  }

  int res = 10000000;
```

```
  for (int i = 0; i < (1<<9); ++i) {
    int cnt = 0;
    for (int j = 0; j < 9; ++j) {
      int x = j/3, y = j%3; st[x][y] = (i>>j)%2;
      if ((i>>j)%2 == 1) ++cnt;
    }
    if (check()) res = min(res, cnt);
  }
  cout << res << endl;
  return 0;
}
```

并查集: 可以约用 O(1) 的时间复杂度 合并两个集合/判断两个数是否在相同集合

P3367 【模板】并查集

并查集 模板题

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

using namespace std;

const int maxn = 2e5 + 5;
int f[maxn];

int fFind(int x) {
  if (f[x] != x) f[x] = fFind(f[x]);
  return f[x];
}

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

  while (m -- ) {
    int op, x, y; cin >> op >> x >> y;
    if (op == 1) {
      int fx = fFind(x), fy = fFind(y);
      if (fx != fy) f[fx] = fy;
    } else {
      if (fFind(x) == fFind(y)) cout << "Y" << endl;
      else cout << "N" << endl;
    }
  }
  return 0;
}
```

P1551 亲戚

并查集 模板题

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

using namespace std;

const int maxn = 5000 + 5;
int f[maxn];

int fFind(int x) {
  if (f[x] != x) f[x] = fFind(f[x]);
  return f[x];
}

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

  while (m -- ) {
    int x, y; cin >> x >> y;
    int fx = fFind(x), fy = fFind(y);
    if (fx != fy) f[fx] = fy;
  }

  while (p -- ) {
    int x, y; cin >> x >> y;
    if (fFind(x) == fFind(y)) cout << "Yes" << endl;
    else cout << "No" << endl;
  }
  return 0;
}
```

P1536 村村通

一共 n 个城市, 全合并起来需要 n-1 次

可以设 cnt = n-1, 之后每合并一次就让 cnt-- 就可以了

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

using namespace std;

const int maxn = 1000 + 5;
int f[maxn];

int fFind(int x) {
  if (f[x] != x) f[x] = fFind(f[x]);
  return f[x];
}
```

```
int n, m;

void solve() {
  for (int i = 1; i <= n; ++i) f[i] = i;
  int cnt = n - 1;
  while (m -- ) {
    int x, y; cin >> x >> y;
    int fx = fFind(x), fy = fFind(y);
    if (fx != fy) { f[fx] = fy; --cnt; }
  }
  cout << cnt << endl;
}

int main()
{
  while (true) {
    cin >> n;
    if (n == 0) break;
    cin >> m;
    solve();
  }
  return 0;
}
```

P1621 集合

先用 埃氏筛 找出所有的质数, 然后枚举比 p 大的质数 i, 找到 a~b 中所有 p 的倍数, 将他们利用并查集合并。

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

using namespace std;

const int maxn = 1e5 + 5;
int f[maxn];
bool st[maxn];

int fFind(int x) {
  if (f[x] != x) f[x] = fFind(f[x]);
  return f[x];
}

int main()
{
  for (int i = 2; i < maxn; ++i) {
    if (!st[i]) {
      for (int j = i+i; j < maxn; j += i) st[j] = true;
    }
  }

  int a, b, p; cin >> a >> b >> p;
```

```
  for (int i = a; i <= b; ++i) f[i] = i;

  int cnt = b - a + 1;
  for (int i = p; i <= b; ++i) {
    if (st[i]) continue;

    int x = i;
    while (x < a) x += i;
    for (int j = x+i; j <= b; j += i) {
      int fx = fFind(x), fj = fFind(j);
      if (fx != fj) f[fx] = fj, --cnt;
    }
  }
  cout << cnt << endl;
  return 0;
}
```

P1929 迷之阶梯

bfs, bfs 中维护的属性为 (pos,k) 两个属性, 代表 后移k步,到达pos位置 这个状态, 最少需要几次移动

当 k==0 时, 说明不是后移的

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

using namespace std;

const int maxn = 200 + 5;
int w[maxn], f[maxn][maxn];
struct node {
  int pos, k;
};

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

  memset(f, -1, sizeof(f));
  queue<node> q; q.push({1, 0}); f[1][0] = 0;
  while (!q.empty()) {
    node u = q.front(); q.pop();
    int pos = u.pos, k = u.k;

    if (pos+1<=n && w[pos+1]==w[pos]+1 && f[pos+1][0]==-1) {
      q.push({pos+1, 0}); f[pos+1][0] = f[pos][k]+1;
    }

    if (pos!=1 && f[pos-1][k+1]==-1) {
      q.push({pos-1, k+1}); f[pos-1][k+1] = f[pos][k]+1;
    }
```

```
      if (k >= 1) {
        int len = (1<<k);
        int up = w[pos] + len;
        for (int j = 1; j <= n; ++j) {
          if (w[j]<=up && f[j][0] == -1) {
            q.push({j, 0}); f[j][0] = f[pos][k]+1;
          }
        }
      }
    }

    cout << f[n][0] << endl;
    return 0;
}
```