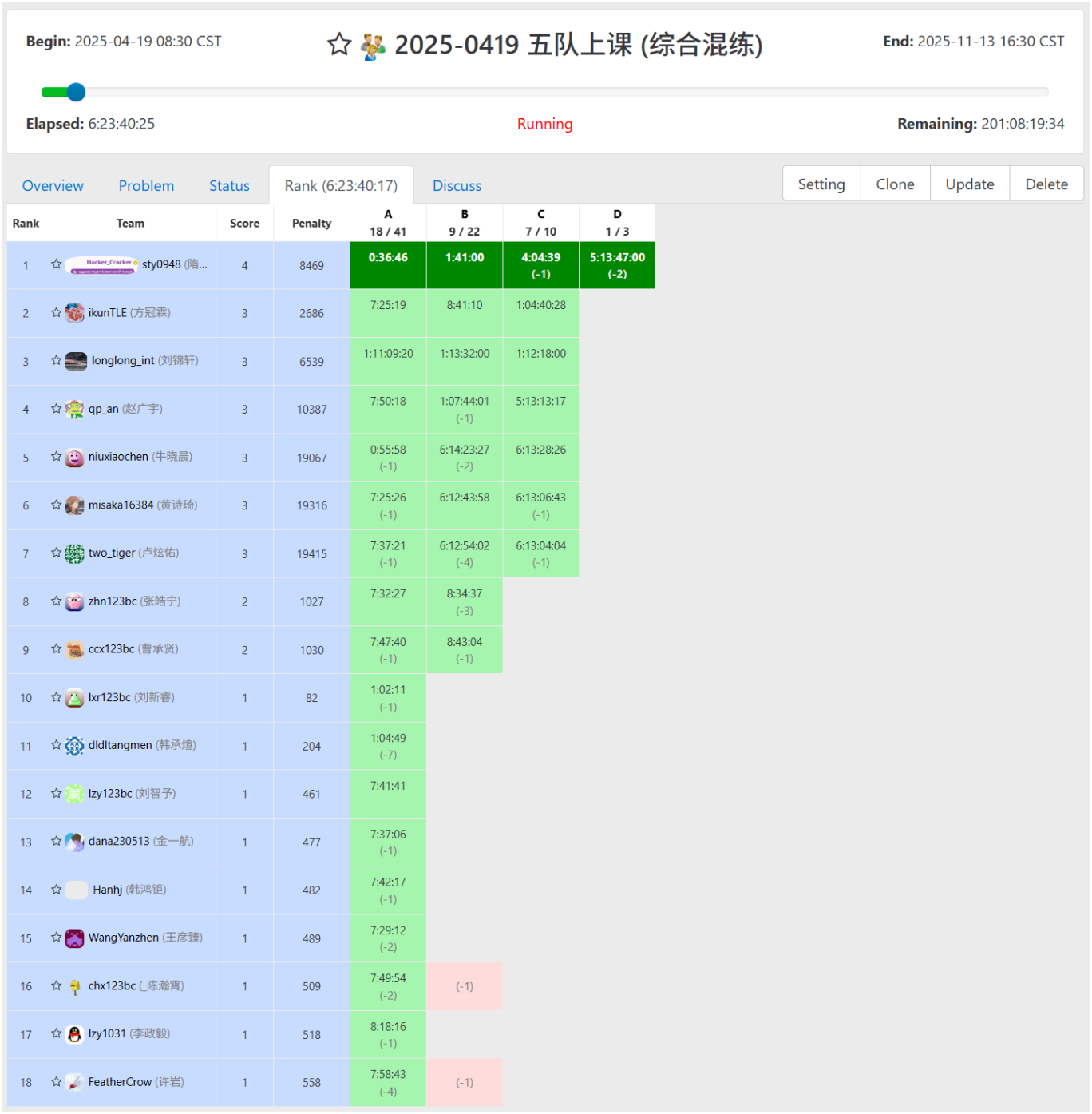
综合混练

人员

隋天翼、刘新睿、牛晓晨、刘锦轩 到课

上周作业检查

上周作业链接: https://vjudge.net/contest/710557



Begin: 2025-04-19 08:30 CST ☆ 2025-0419 五队上课 (综合混练) **End:** 2025-11-13 16:30 CST

Elapsed: 6:23:40:25 Running **Remaining:** 201:08:19:34

Overview Problem Status Rank (6:23:40:17) Discuss Setting Clone Update Delete

Rank	Team	Score	Penalty	A 18 / 41	B 9 / 22	C 7 / 10	D 1 / 3
1	☆ sty0948 (隋...	4	8469	0:36:46	1:41:00	4:04:39 (-1)	5:13:47:00 (-2)
2	☆ ikunTLE (方冠霖)	3	2686	7:25:19	8:41:10	1:04:40:28	
3	☆ longlong_int (刘锦轩)	3	6539	1:11:09:20	1:13:32:00	1:12:18:00	
4	☆ qp_an (赵广宇)	3	10387	7:50:18	1:07:44:01 (-1)	5:13:13:17	
5	☆ niuxiaochen (牛晓晨)	3	19067	0:55:58 (-1)	6:14:23:27 (-2)	6:13:28:26	
6	☆ misaka16384 (黄诗琦)	3	19316	7:25:26 (-1)	6:12:43:58	6:13:06:43 (-1)	
7	☆ two_tiger (卢炫佑)	3	19415	7:37:21 (-1)	6:12:54:02 (-4)	6:13:04:04 (-1)	
8	☆ zhn123bc (张皓宁)	2	1027	7:32:27	8:34:37 (-3)		
9	☆ ccx123bc (曹承贤)	2	1030	7:47:40 (-1)	8:43:04 (-1)		
10	☆ lxr123bc (刘新睿)	1	82	1:02:11 (-1)			
11	☆ dldltangmen (韩承煊)	1	204	1:04:49 (-7)			
12	☆ lzy123bc (刘智予)	1	461	7:41:41			
13	☆ dana230513 (金一航)	1	477	7:37:06 (-1)			
14	☆ Hanhj (韩鸿钜)	1	482	7:42:17 (-1)			
15	☆ WangYanzhen (王彦臻)	1	489	7:29:12 (-2)			
16	☆ chx123bc (_陈瀚霄)	1	509	7:49:54 (-2)	(-1)		
17	☆ lzy1031 (李政毅)	1	518	8:18:16 (-1)			
18	☆ FeatherCrow (许岩)	1	558	7:58:43 (-4)	(-1)		

作业

https://vjudge.net/contest/712438 (课上讲了 A ~ C 这些题, 课后作业是 D E 题)

课堂表现

同学们今天整体上课表现都很不错, 就是课下要多花时间补题, 作业也一定要多花时间想一想做一做, 不能老是不做作业等老师讲。

课堂内容

CF1454E Number of Simple Paths

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

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
vector<int> vec[maxn];
int deg[maxn];
set<int> s;
int cnt;

void dfs(int u, int fa) {
  ++cnt;
  for (int i : vec[u]) {
    if (i==fa || s.count(i)) continue;
    dfs(i, u);
  }
}

void solve() {
  int n; cin >> n;
  for (int i = 0; i <= n+2; ++i) vec[i].clear(), deg[i] = 0;

  for (int i = 1; i <= n; ++i) {
    int u, v; cin >> u >> v;
    vec[u].push_back(v), vec[v].push_back(u);
    deg[u]++, deg[v]++;
  }

  s.clear(); for (int i = 1; i <= n; ++i) s.insert(i);

  queue<int> q;
  for (int i = 1; i <= n; ++i) {
    if (deg[i] == 1) q.push(i), s.erase(i);
  }
  while (!q.empty()) {
    int u = q.front(); q.pop();
    for (int i : vec[u]) {
      --deg[i]; if (deg[i] == 1) q.push(i), s.erase(i);
    }
  }

  LL res = (LL)n * (n-1);
```

```
  for (int i : s) {
    cnt = 0; dfs(i, -1); res -= (LL)cnt * (cnt-1) / 2;
  }

//  cout << "---------------------- ";
  cout << res << endl;
}

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

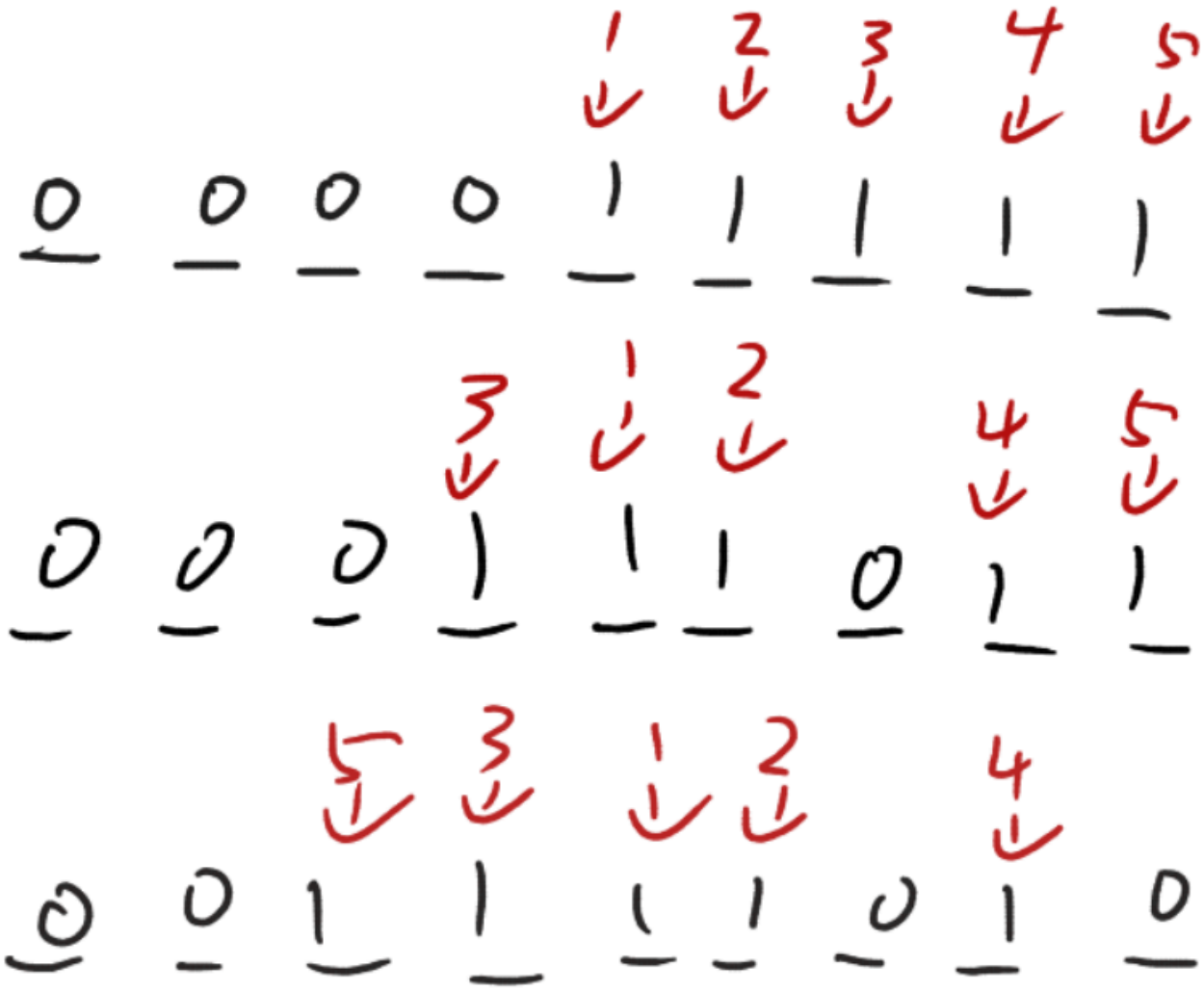
CF1272D Remove One Element

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

using namespace std;

const int maxn = 2e5 + 5;
int w[maxn];
int f1[maxn], f2[maxn];

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

  int res = 0;
  for (int i = 1; i <= n; ++i) {
    if (w[i] > w[i-1]) f1[i] = f1[i-1]+1;
    else f1[i] = 1;
    res = max(res, f1[i]);
  }

  for (int i = n; i >= 1; --i) {
    if (w[i] < w[i+1]) f2[i] = f2[i+1]+1;
    else f2[i] = 1;
  }

  for (int i = 2; i <= n-1; ++i) {
    if (w[i-1] < w[i+1]) res = max(res, f1[i-1] + f2[i+1]);
  }
  cout << res << endl;
  return 0;
}
```

CF1551E Fixed Points

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

using namespace std;

const int maxn = 2000 + 5;
int w[maxn], f[maxn][maxn];

void solve() {
  int n, k; cin >> n >> k;
  for (int i = 0; i <= n+2; ++i) {
    for (int j = 0; j <= n+2; ++j) f[i][j] = 0;
  }

  for (int i = 1; i <= n; ++i) cin >> w[i];
  for (int i = 1; i <= n; ++i) {
    f[i][0] = f[i-1][0] + (w[i]==i);
    for (int j = 1; j < i; ++j) {
      f[i][j] = max(f[i-1][j-1], f[i-1][j] + (w[i]==i-j));
    }
  }

//  cout << "----------------------------- ";
  for (int i = 0; i < n; ++i) {
    if (f[n][i] >= k) { cout << i << endl; return; }
  }
  cout << -1 << endl;
}

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

CF1288E Messenger Simulator



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

using namespace std;

const int N = 3e5 + 5;
int up[N], down[N], f[N];

int tr[N*2];
int lowbit(int x) { return x&(-x); }
void update(int x, int k) {
  while (x < N*2) tr[x] += k, x += lowbit(x);
}
int query(int x) {
  int res = 0;
  while (x) res += tr[x], x -= lowbit(x);
  return res;
}

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

```
    f[i] = N+i, update(f[i], 1);
  }

  for (int i = 1; i <= m; ++i) {
    int x; cin >> x;
    down[x] = max(down[x], query(f[x])), up[x] = 1;
    update(f[x], -1);
    f[x] = N-i, update(f[x], 1);
  }

  for (int i = 1; i <= n; ++i) down[i] = max(down[i], query(f[i]));

  for (int i = 1; i <= n; ++i) cout << up[i] << " " << down[i] << endl;
  return 0;
}
```