

# 带权并查集

---

## 人员

刘佳赫、左子毅、杨洋、于珈浩、朱奕鸣 到课

## 作业

<https://www.luogu.com.cn/contest/212458>

A、B 为课上习题，C、D 为之前习题，E、F 为课后作业

## 课堂表现

同学们课上听讲整体都很认真，希望同学们继续保持，课下也要积极补题。

## 课堂内容

### P2024 [NOI2001] 食物链

$f[i]$ : 维护  $i$  的父亲

$d[i]$ : 维护  $i$  与  $i$  的父亲 之间的距离

路径压缩代码:

```
int fFind(int x) {
    if (f[x] != x) {
        int p = f[x];
        f[x] = fFind(f[x]);
        d[x] = (d[x] + d[p]) % 3;
    }
}
```

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

using namespace std;

const int maxn = 5e4 + 5;
const int mod = 3;
int f[maxn], d[maxn];

int fFind(int x) {
    if (f[x] != x) {
        int p = f[x];
        f[x] = fFind(f[x]);
        d[x] = (d[x] + d[p]) % mod;
    }
}
```

```

    return f[x];
}

int main()
{
    int n, k, res = 0; cin >> n >> k;
    for (int i = 1; i <= n; ++i) f[i] = i;

    while (k -- ) {
        int op, x, y; cin >> op >> x >> y;
        if (x>n || y>n) { ++res; continue; }

        int p = fFind(x), q = fFind(y);
        if (p == q) {
            if (op==1 && d[x]!=d[y]) ++res;
            if (op==2 && d[x]!=(d[y]+1)%mod) ++res;
        } else {
            f[p] = q, d[p] = (d[y]+(op-1)-d[x]+mod) % mod;
        }
    }

    cout << res << endl;
    return 0;
}

```

## U492914 区间覆盖

1. 按照左端点从小到大排序
2. 从左到右扫一遍，如果这个区间的左端点  $\leq$  限定值，就说明这个可选，对右端点进行取 max 操作 否则，说明不可直接选，先选前面的区间，再尝试选这个区间

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e5 + 5;
struct node {
    int l, r;
} w[maxn];

bool cmp(node p, node q) {
    return p.l < q.l;
}

int main()
{
    int s, t; cin >> s >> t;
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i].l >> w[i].r;
    sort(w+1, w+n+1, cmp);
}

```

```
int cnt = 0, last = s, r = -1e9;
for (int i = 1; i <= n; ++i) {
    if (w[i].l <= last) {
        r = max(r, w[i].r);
        if (r >= t) {
            ++cnt; cout << cnt << endl; return 0;
        }
    } else {
        ++cnt; last = r; r = -1e9;
        if (w[i].l <= last) {
            r = max(r, w[i].r);
            if (r >= t) {
                ++cnt; cout << cnt << endl; return 0;
            }
        } else {
            cout << -1 << endl; return 0;
        }
    }
}

cout << -1 << endl;
return 0;
}
```