

杂题混练

人员

穆鹏宇、梁钰涵、蔡云翔、程晟泰 到课

上周作业检查

Begin: 2025-01-18 09:30 CST

☆ 2025-0118 ~ 0119 三队上课

End: 2025-08-14 17:30 CST

6:23:04:27

Elapsed: 6:23:04:27

Running

Remaining: 201:08:55:32

Overview

Problem

Status

Rank (6:23:04:26)

Discuss

Setting

Clone

Update

Delete

Rank	Team	Score	Penalty	A 1 / 3
1	☆ syzliangyuhan (梁钰涵)	1	546	8:46:05 (-1)
2	☆ ssine233 (穆鹏宇)	0	0	(-1)

作业

https://vjudge.net/contest/688559

课堂表现

今天题目整体会复杂一些, 同学们课上基本只完成了 A 题, 课下要再完成一下 B、C 题

课堂内容

Gym - 103428G Shinyruo and KFC

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

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
const int mod = 998244353;
int fac[maxn], inv_fac[maxn];

int qmod(int a, int k) {
    int res = 1;
    while (k) {
        if (k&1) res = (LL)res*a % mod;
        a = (LL)a*a % mod;
        k >>= 1;
    }
    return res;
}
```

```

}

void init() {
    fac[0] = inv_fac[0] = 1;
    for (int i = 1; i < maxn; ++i) {
        fac[i] = 1LL*fac[i-1]*i % mod; inv_fac[i] = qmod(fac[i], mod-2);
    }
}

int C(int n, int m) {
    if (n<m) return 0;
    return 1LL * fac[n] * inv_fac[m] % mod * inv_fac[n-m] % mod;
}

int f[maxn];
struct node {
    int val, cnt;
};

int main()
{
    init();

    int n, m; cin >> n >> m;
    int maxx = 0;
    for (int i = 1; i <= n; ++i) {
        int x; cin >> x; f[x]++; maxx = max(maxx, x);
    }

    vector<node> vec;
    for (int i = 1; i < maxn; ++i) {
        if (f[i]) vec.push_back({i, f[i]});
    }

    for (int i = 1; i <= m; ++i) {
        if (i < maxx) { cout << 0 << endl; continue; }

        int res = 1;
        for (node it : vec) res = 1LL * res * qmod(C(i,it.val), it.cnt) % mod;
        cout << res << endl;
    }
    return 0;
}

```

UVA1599 理想路径 Ideal Path

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e5 + 5;

```

```
struct node {
    int to, value;
};
vector<node> vec[maxn];
int dis[maxn];
int n;
bool st[maxn];

void bfs() {
    memset(dis, -1, sizeof(dis));
    queue<int> q; q.push(n); dis[n] = 0;
    while (!q.empty()) {
        int u = q.front(); q.pop();
        for (node it : vec[u]) {
            if (dis[it.to] == -1) q.push(it.to), dis[it.to] = dis[u]+1;
        }
    }
}

void bfs2() {
    queue<int> q; q.push(1);
    bool flag = false;
    while (!q.empty()) {
        vector<int> vv;
        while (!q.empty()) vv.push_back(q.front()), q.pop();

        int minn = 1e9 + 5;
        for (int i : vv) {
            for (node it : vec[i]) {
                if (dis[it.to]==dis[i]-1) minn = min(minn, it.value);
            }
        }

        if (minn == 1e9+5) break;
        if (flag == true) cout << " ";
        flag = true;
        cout << minn;

        for (int i : vv) {
            for (node it : vec[i]) {
                if (dis[it.to]==dis[i]-1 && it.value==minn && !st[it.to]) {
                    q.push(it.to); st[it.to] = true;
                }
            }
        }
    }
    cout << endl;
}

int main()
{
    int m;
    while (cin >> n >> m) {
        for (int i = 1; i <= n; ++i) vec[i].clear(), st[i] = false;
```

```

while (m -- ) {
    int u, v, w; cin >> u >> v >> w;
    vec[u].push_back({v,w}), vec[v].push_back({u,w});
}

bfs();
cout << dis[1] << endl;
bfs2();
}
return 0;
}

```

P1712 [NOI2016] 区间

```

#include <bits/stdc++.h>

using namespace std;

vector<int> ys;
int yFind(int x) {
    return lower_bound(ys.begin(), ys.end(), x) - ys.begin();
}

const int N = 5e5 + 5, M = 1e6 + 5;
const int inf = 0x3f3f3f3f;

struct Info {
    int l, r, d;
    bool operator < (const Info& p) const {
        return d < p.d;
    }
} w[N];

struct node {
    int l, r;
    int maxx, add;
} tr[M<<2];

void pushup(int u) {
    tr[u].maxx = max(tr[u<<1].maxx, tr[u<<1|1].maxx);
}

void pushdown(int u) {
    if (tr[u].add) {
        tr[u<<1].add += tr[u].add, tr[u<<1].maxx += tr[u].add;
        tr[u<<1|1].add += tr[u].add, tr[u<<1|1].maxx += tr[u].add;
        tr[u].add = 0;
    }
}

void build(int u, int l, int r) {
    tr[u] = {l, r};
    if (l != r) {

```

```

        int mid = (l + r) >> 1;
        build(u<<1, l, mid); build(u<<1|1, mid+1, r);
    }
}

void modify(int u, int l, int r, int k) {
    if (tr[u].l >= l && tr[u].r <= r) {
        tr[u].add += k, tr[u].maxx += k;
        return;
    }
    pushdown(u);
    int mid = (tr[u].l + tr[u].r) >> 1;
    if (l <= mid) modify(u<<1, l, r, k);
    if (r > mid) modify(u<<1|1, l, r, k);
    pushup(u);
}

int query(int u, int l, int r) {
    if (tr[u].l >= l && tr[u].r <= r) return tr[u].maxx;
    pushdown(u);
    int mid = (tr[u].l + tr[u].r) >> 1;
    int res = 0;
    if (l <= mid) res = query(u<<1, l, r);
    if (r > mid) res = max(res, query(u<<1|1, l, r));
    return res;
}

int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);

    int n, m; cin >> n >> m;
    for (int i = 1; i <= n; ++i) {
        cin >> w[i].l >> w[i].r;
        w[i].d = w[i].r - w[i].l;
        ys.push_back(w[i].l), ys.push_back(w[i].r);
    }
    sort(ys.begin(), ys.end());
    ys.erase(unique(ys.begin(), ys.end()), ys.end());

    sort(w + 1, w + n + 1);
    build(1, 1, M - 1);
    int res = inf;
    for (int i = 1, j = 1; j <= n; ++j) {
        int jL = yFind(w[j].l) + 1, jR = yFind(w[j].r) + 1;
        modify(1, jL, jR, 1);
        while (i <= j && query(1, 1, M-1) >= m) {
            res = min(res, w[j].d - w[i].d);
            int iL = yFind(w[i].l) + 1, iR = yFind(w[i].r) + 1;
            modify(1, iL, iR, -1);
            ++i;
        }
    }

    if (res == inf) cout << -1 << endl;
}

```

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