

并查集

人员

赵熙羽、杨瑾硕、谢亚锴、于子珈、隋天乙、刘闯速、牛同泽、孙靖轲 到课

上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/793>

2025-0914 周日10:30 (综合练习)

刷新

#	用户名	姓名	编程分	时间	A	B	C	D
1	yuzjia1	于子珈	400	8310	100	100	100	100
2	zhaoxiyu	赵熙羽	400	15709	100	100	100	100
3	sunjingke	孙靖轲	400	15821	100	100	100	100
4	xieyakai	谢亚锴	358	14447	100	100	100	58
5	qinxiansen	秦显森	358	14698	100	100	100	58
6	siyunxin	司云心	300	15065	100	100	100	
7	chenluoran	陈洛冉	300	15721	100	100	100	
8	yangjinsuo	杨瑾硕	300	15905	100	100	100	
9	niutongze	牛同泽	250	14230	100	100		50
10	suitianyi	隋天乙	206	606	100	91	15	
11	liuchuangsu	刘闯速	200	1068	100		100	
12	gezhenran	葛真然	200	14084	100	100		
13	chengzihao	程梓豪	200	14375	100	100		

本周作业

<https://cppoj.kids123code.com/contest/846> (课上讲了 A ~ E 题, 课后作业是 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

AT_abc308_d [ABC308D] Snuke Maze (上周作业)

判断从 左上角 到 右下角 有没有一条只由 snuke 组成的路径

从左上角往右下搜索, 保证每个点只访问一次

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

using namespace std;

map<char, char> mp;
const int maxn = 500 + 5;
```

```

char s[maxn][maxn];
int n, m;
int dx[] = {-1, 1, 0, 0}, dy[] = {0, 0, -1, 1};
bool st[maxn][maxn];

bool dfs(int x, int y) {
    if (st[x][y]) return false;
    st[x][y] = true;
    if (x==n && y==m) return true;

    for (int i = 0; i < 4; ++i) {
        int nx = x+dx[i], ny = y+dy[i];
        if (s[nx][ny]==mp[s[x][y]] && dfs(nx,ny)) return true;
    }
    return false;
}

int main()
{
    mp['s'] = 'n', mp['n'] = 'u', mp['u'] = 'k', mp['k'] = 'e', mp['e'] = 's';

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

    if (s[1][1]=='s' && dfs(1,1)) cout << "Yes" << endl;
    else cout << "No" << 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

【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;
}
```

亲戚

并查集 模板题

```
#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;
}
```

村村通

一共 n 个城市, 全合并起来需要 $n-1$ 次

可以设 $\text{cnt} = n-1$, 之后每合并一次就让 $\text{cnt}--$ 就可以了

```
#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;
}
```

素数个数

埃氏筛 模板, 当 $f[i]=\text{true}$ 时, 说明 i 是合数

原理: 所有质数的倍数都是合数

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

using namespace std;
```

```

const int maxn = 1e8 + 5;
bool f[maxn];

int main()
{
    int n; cin >> n;
    for (int i = 2; i <= n; ++i) {
        if (f[i]) continue;
        for (int j = i+i; j <= n; j += i) f[j] = true;
    }

    int res = 0;
    for (int i = 2; i <= n; ++i) {
        if (!f[i]) ++res;
    }
    cout << res << endl;
    return 0;
}

```

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 $a \sim b$ 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;
}
```