

黑白染色

人员

杨咏丞、赵书梵、李瑞涵 到课, 柳力玮、高健桓、苑钊 线上

上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/2586>

#	用户名	姓名	编程分	时间	A	B	C	D	E	F	G
1	zhaoshufan	赵书梵	700	8131	100	100	100	100	100	100	100
2	liuliwei	柳力玮	700	10361	100	100	100	100	100	100	100
3	yangyongcheng	杨咏丞	500	8596	100	100	100	100	100		
4	gaojianhuan	高健桓	433	11296	100	100	100	100	33		
5	yuanzhao	苑钊	423	6757	100	100	100	100	23		
6	jiangshuzhang	蒋叔璋	400	11037	100	100	100	100			
7	tianxinyi	田心一	400	11353	100	100	100	100			

本周作业

<https://cppoj.kids123code.com/contest/2686> (课上讲了 A ~ D 题, 课后作业是 E 题必做)

课堂表现

今天课上讲了黑白染色的内容, 同学们课上整体掌握都比较好。

课堂内容

[GESP202503 六级] 树上漫步

黑白染色, 把所有点分成黑点和白点, 要求相邻的点颜色不能相同

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

using namespace std;

const int maxn = 2e5 + 5;
vector<int> vec[maxn];
int f[maxn];

void dfs(int u, int val) {
    if (f[u] != -1) return;
    f[u] = val;
    for (int i : vec[u]) dfs(i, 1-val);
```

```

}

int main()
{
    memset(f, -1, sizeof(f));
    int n; cin >> n;
    for (int i = 1; i <= n-1; ++i) {
        int a, b; cin >> a >> b;
        vec[a].push_back(b), vec[b].push_back(a);
    }

    dfs(1, 0);

    int cnt0 = 0, cnt1 = 0;
    for (int i = 1; i <= n; ++i) {
        if (!f[i]) ++cnt0;
        else ++cnt1;
    }

    for (int i = 1; i <= n; ++i) {
        if (!f[i]) cout << cnt0 << " ";
        else cout << cnt1 << " ";
    }
    cout << endl;
    return 0;
}

```

[GESP202403 七级] 交流问题

给每个联通块进行黑白染色，并计算每个联通块中有几个黑点，有几个白点

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 2e5 + 5;
vector<int> vec[maxn];
int f[maxn], cnt0, cnt1;

void dfs(int u, int val) {
    if (f[u] != -1) return;
    f[u] = val;
    if (!val) ++cnt0;
    else ++cnt1;

    for (int i : vec[u]) dfs(i, 1-val);
}

int main()
{
    memset(f, -1, sizeof(f));

```

```

int n, m; cin >> n >> m;
while (m -- ) {
    int a, b; cin >> a >> b;
    vec[a].push_back(b), vec[b].push_back(a);
}

int res1 = 0, res2 = 0;
for (int i = 1; i <= n; ++i) {
    if (f[i] == -1) {
        cnt0 = 0, cnt1 = 0;
        dfs(i, 0);
        res1 += min(cnt0, cnt1), res2 += max(cnt0, cnt1);
    }
}
cout << res1 << " " << res2 << endl;
return 0;
}

```

[ROI 2013 Day1] 乌拉尔冰球赛

给每个联通块进行黑白染色, 把每个联通块中颜色多的点找出来存下来

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e5 + 5;
vector<int> vec[maxn];
int f[maxn], cnt0 = 0, cnt1 = 0;
vector<int> v0, v1;

void dfs(int u, int val) {
    if (f[u] != -1) return;
    f[u] = val;

    if (!val) ++cnt0, v0.push_back(u);
    else ++cnt1, v1.push_back(u);

    for (int i : vec[u]) dfs(i, 1-val);
}

int main()
{
    memset(f, -1, sizeof(f));
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) {
        int a, b; cin >> a >> b;
        vec[a].push_back(b), vec[b].push_back(a);
    }
    int K; cin >> K;
}

```

```

vector<int> ans;
for (int i = 1; i <= n; ++i) {
    if (f[i] == -1) {
        cnt0 = 0, cnt1 = 0; v0.clear(), v1.clear();
        dfs(i, 0);
        if (cnt0 > cnt1) {
            for (int i : v0) ans.push_back(i);
        } else {
            for (int i : v1) ans.push_back(i);
        }
    }
}

if ((int)ans.size() < K) cout << 0 << endl;
else {
    for (int i = 0; i < K; ++i)
        cout << ans[i] << " ";
    cout << endl;
}
return 0;
}

```

封锁阳光大学

黑白染色, 判定某个联通块中是否有奇环的冲突

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 2e5 + 5;
vector<int> vec[maxn];
int f[maxn], cnt0, cnt1;
bool flag;

void dfs(int u, int val) {
    if (!flag) return;
    if (f[u] != -1) {
        if (f[u] != val) flag = false;
        return;
    }
    f[u] = val;

    if (!val) ++cnt0;
    else ++cnt1;

    for (int i : vec[u]) dfs(i, 1-val);
}

int main()
{

```

```
memset(f, -1, sizeof(f));
int n, m; cin >> n >> m;
while (m -- ) {
    int a, b; cin >> a >> b;
    vec[a].push_back(b), vec[b].push_back(a);
}

int res = 0;
for (int i = 1; i <= n; ++i) {
    if (f[i] == -1) {
        cnt0 = 0, cnt1 = 0; flag = true;
        dfs(i, 0);
        if (!flag) { cout << "Impossible" << endl; return 0; }
        res += min(cnt0, cnt1);
    }
}
cout << res << endl;
return 0;
}
```