

Tên: Thái Quốc Bảo
MSSV: 3123410031 – STT: 07
Welsh-Powel

```

// 1. Đọc dữ liệu
#include <iostream>
#include <vector>
#include <algorithm>
#define maxN 10000

using namespace std;

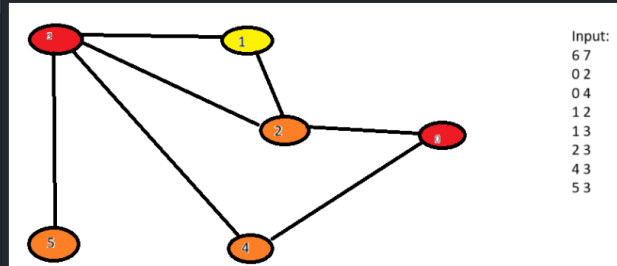
int n, m;
vector<int> v[maxN];
int color[maxN] = {0};

// Hàm nhập đồ thị
Tabnine | Edit | Test | Explain | Document
void input(vector<int> *v, const int &n, const int &m)
{
    for (int i = 0; i < m; i++)
    {
        int u, w;
        cin >> u >> w;
        v[u].push_back(w);
        v[w].push_back(u);
    }
}

// Kiểm tra xem màu có hợp lệ với đỉnh u không
Tabnine | Edit | Test | Explain | Document
bool check(int u, int c)
{
    for (int tmp : v[u])
    {
        if (color[tmp] == c)
            return false;
    }
    return true;
}

```

Download | ImageProxy



// Tô màu cho đỉnh nếu hợp lệ

Tabnine | Edit | Test | Explain | Document

int ToMau(int u, int mau)

```
{
    if (!color[u] && check(u, mau))
    {
        color[u] = mau;
        return 1;
    }
    return 0;
}
```

// Hàm sắp xếp đỉnh theo bậc giảm dần

Tabnine | Edit | Test | Explain | Document

vector<int> sortVerticesByDegree()

```
{
    vector<pair<int, int>> degree;
    for (int i = 0; i < n; i++)
    {
        degree.push_back({v[i].size(), i});
    }
    sort(degree.rbegin(), degree.rend());

    vector<int> sortedVertices;
    for (auto &p : degree)
    {
        sortedVertices.push_back(p.second);
    }
    return sortedVertices;
}
```

Tabnine | Edit | Test | Explain | Document

```
int main()
{
    cin >> n >> m;
    input(v, n, m);

    // Sắp xếp đỉnh theo bậc giảm dần
    vector<int> sortedVertices = sortVerticesByDegree();

    int colored = 0, mau = 1;
    while (colored < n)
    {
        for (int i : sortedVertices)
        {
            colored += ToMau(i, mau);
        }
        mau++;
    }

    output();
    return 0;
}
```

DSatur

```
#include <iostream>
#include <vector>
#include <set>
using namespace std;
```

Tabnine | Edit | Test | Explain | Document

```
int getMaxSat(const vector<set<int>> &g, const vector<int> &c, const vector<int> &s)
{
    int max_s = -1, max_d = -1, v = -1;
    for (int i = 0; i < g.size(); i++)
    {
        if (c[i] == -1 && (s[i] > max_s || (s[i] == max_s && g[i].size() > max_d)))
        {
            max_s = s[i];
            max_d = g[i].size();
            v = i;
        }
    }
    return v;
}
```

Tabnine | Edit | Test | Explain | Document

```
void dsatur(const vector<set<int>> &g)
{
    int n = g.size();
    vector<int> c(n, -1), s(n, 0);

    for (int i = 0; i < n; i++)
    {
        int v = getMaxSat(g, c, s);

        set<int> used;
        for (int nb : g[v])
            if (c[nb] != -1)
                used.insert(c[nb]);

        int clr = 0;
        while (used.count(clr))
            clr++;
        c[v] = clr;
    }
}
```

```

        for (int nb : g[v])
        {
            if (c[nb] == -1)
            {
                set<int> u;
                for (int n2 : g[nb])
                    if (c[n2] != -1)
                        u.insert(c[n2]);
                s[nb] = u.size();
            }
        }
    }

    for (int i = 0; i < n; i++)
        cout << "Dinh " << i << " mau: " << c[i] << endl;
}

```

Tabnine | Edit | Test | Explain | Document

```

int main()
{
    int n, m;
    cin >> n >> m;
    vector<set<int>> g(n);

    for (int i = 0; i < m; i++)
    {
        int u, v;
        cin >> u >> v;
        g[u].insert(v);
        g[v].insert(u);
    }

    dsatur(g);
    return 0;
}

```