

图的遍历

人员

田心一、纪博涵、李瑞涵、杨咏丞 到课, 蒋叔璋、初锦阳、柳力玮、苑钊 线上

上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/2332>

🏠 比赛概况

📋 题目列表

📋 选择题列表

📄 提交记录

★ 实时榜单

★ 选择题排行榜

王向东老师周日三点半C++存图搜图

🔄 刷新

#	用户名	姓名	编程分	时间	A	B	C	D	E	F
1	liuliwei	柳力玮	600	5396	100	100	100	100	100	100
2	liuchenxi	刘宸熙	500	2970	100	100	100	100	100	
3	tianxinyi	田心一	500	3886	100	100	100	100	100	0
4	yangyongcheng	杨咏丞	500	5276	100	100	100	100	100	
5	chujinyang	初锦阳	500	6110	100	100	100	100	100	
6	li Ruihan	李瑞涵	500	6347	100	100	100	100	100	
7	jibohan	纪博涵	500	6567	100	100	100	100	100	
8	zhaoshufan	赵书梵	400	1999	100	100	100	100		
9	yuanzhao	苑钊	400	4681	100	100	100	100		
10	jiangshuzhang	蒋叔璋	300	5956	100	100	100	0		
11	gaojianhuan	高健桓	0	0	0					

本周作业

<https://cppoj.kids123code.com/contest/2456> (课上讲了 A ~ E 题, 课后作业是 F 题)

课堂表现

今天主要针对图的 dfs 和 bfs 遍历进行了讲解, D 题是个图的 bfs 找最短路, 同学们普遍做的不太熟练, 一定要课下再多写几遍。

课堂内容

【深基18.例3】查找文献 (上周作业)

以 1 为起点进行 dfs 搜索 + bfs 搜索

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

using namespace std;

const int maxn = 1e5 + 5;
vector<int> vec[maxn];
bool vis[maxn];
```

```

void dfs(int u) {
    if (vis[u]) return;
    vis[u] = true;
    cout << u << " ";

    for (int i : vec[u]) dfs(i);
}

void bfs(int x) {
    memset(vis, false, sizeof(vis));
    queue<int> q; q.push(x); vis[x] = true; cout << x << " ";
    while (!q.empty()) {
        int u = q.front(); q.pop();
        for (int i : vec[u]) {
            if (vis[i]) continue;
            q.push(i); vis[i] = true; cout << i << " ";
        }
    }
    cout << endl;
}

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

    for (int i = 1; i <= n; ++i) sort(vec[i].begin(), vec[i].end());

    dfs(1);
    cout << endl;

    bfs(1);
    return 0;
}

```

高手去散步

用 vector vec[maxn] 的方式存带权图

以任意点为起点, 往后能搜就搜, 把所有搜的情况找全, 找出一条最大路径

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 20 + 5;
struct node {

```

```

    int to, value;
};
vector<node> vec[maxn];
int res = 0;
bool st[maxn];

void dfs(int u, int len) {
    if (st[u]) return;

    res = max(res, len);
    st[u] = true;
    for (node it : vec[u]) dfs(it.to, len+it.value);
    st[u] = false;
}

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

    for (int i = 1; i <= n; i++) dfs(i, 0);

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

```

Ladder Takahashi

点的值域比较大, 所以可以用 `map<int,vector>` 来存图搜图即可

```

#include <bits/stdc++.h>

using namespace std;

map<int, vector<int>> mp;
set<int> st;

int res = 0;

void dfs(int u) {
    if (st.count(u)) return;
    st.insert(u);

    res = max(res, u);
    for (int i : mp[u]) dfs(i);
}

```

```
int main()
{
    int n; cin >> n;
    while (n -- ) {
        int a, b; cin >> a >> b;
        mp[a].push_back(b), mp[b].push_back(a);
    }

    dfs(1);

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

Count Connected Components

求一个图里面有几个联通块, 方法跟前面一样, 不过是从二维数组变到图上来求

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

using namespace std;

const int maxn = 100 + 5;
vector<int> vec[maxn];
bool st[maxn];

void dfs(int u) {
    if (st[u]) return;
    st[u] = true;

    for (int i : vec[u]) dfs(i);
}

int main()
{
    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 (st[i]) continue;
        ++res; dfs(i);
    }
    cout << res << endl;
    return 0;
}
```

[蓝桥杯青少年组省赛 2022] 路线

以点 n 为起点, 做 bfs 搜索, 求到每个点的最短路是多少

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

using namespace std;

const int maxn = 100 + 5;
vector<int> vec[maxn];
int dis[maxn];

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

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

    for (int i = 1; i <= n-1; ++i) cout << dis[i] << " ";
    cout << endl;
    return 0;
}
```

[GESP202312 六级] 工作沟通

建有向图, 然后枚举点 i , 从 $n-1$ 枚举到 0 , 看哪个点能把题目要求的 m 个点都给访问到。

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

using namespace std;

const int maxn = 300 + 5;
vector<int> vec[maxn];
bool st[maxn];

void dfs(int u) {
    if (st[u]) return;
    st[u] = true;
```

```
    for (int i : vec[u]) dfs(i);
}

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n-1; ++i) {
        int x; cin >> x; vec[x].push_back(i);
    }

    int m; cin >> m;
    while (m -- ) {
        vector<int> alls;

        int c; cin >> c;
        while (c -- ) {
            int x; cin >> x; alls.push_back(x);
        }

        for (int i = n-1; i >= 0; --i) {
            memset(st, false, sizeof(st));
            dfs(i);

            bool flag = true;
            for (int j : alls) {
                if (!st[j]) flag = false;
            }
            if (flag) { cout << i << endl; break; }
        }
    }
    return 0;
}
```