

综合练习

人员

柳力玮、蒋叔璋、赵书梵 到课, 田心一、杨咏丞、苑钊、高健桓 线上

上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/2456>

王向东老师周日三点半C++图的遍历

#	用户名	姓名	编程分	时间	A	B	C	D	E	F
1	zhaoshufan	赵书梵	600	9359	100	100	100	100	100	100
2	liulwei	柳力玮	600	10290	100	100	100	100	100	100
3	tianxinyi	田心一	575	9643	100	100	100	100	100	75
4	yuanzhao	苑钊	567	6939	100	100	100	100	100	67
5	yangyongcheng	杨咏丞	400	6540	100	100	100	100	0	
6	jiangshuzhang	蒋叔璋	400	8533	100	100	100	100		
7	chujin yang	初锦阳	400	9202	100	100	100	100		
8	jibohan	纪博涵	400	9394	100	100	100	100		
9	liruihan	李瑞涵	400	9841	100	100	100	100		

说点什么

本周作业

<https://cppoj.kids123code.com/contest/2586> (课上讲了 A ~ E 题, 课后作业是 E 题必做, F G 选做)

课堂表现

今天主要针对图进行了一些 dfs 搜索和 bfs 搜索的练习, 能看出来大部分同学对于搜索掌握的还是不熟练, 基本都是老师写了才写出来, 课下需要多复习复习。

课堂内容

Path Graph? (上周作业)

一条链的要求:

1. n 个点, n-1 条边
2. 连通
3. 每个点度数不能 $>= 3$

```
#include <bits/stdc++.h>
using namespace std;
```

```

const int maxn = 2e5 + 5;
int deg[maxn];
vector<int> vec[maxn];
bool vis[maxn];

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

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

int main()
{
    int n, m; cin >> n >> m;
    bool flag = true;
    for (int i = 1; i <= m; ++i) {
        int a, b; cin >> a >> b; ++deg[a], ++deg[b];
        if (deg[a]>=3 || deg[b]>=3) flag = false;
        vec[a].push_back(b), vec[b].push_back(a);
    }

    if (m!=n-1 || !flag) { cout << "No" << endl; return 0; }

    dfs(1);

    for (int i = 1; i <= n; ++i) {
        if (!vis[i]) { cout << "No" << endl; return 0; }
    }
    cout << "Yes" << endl;
    return 0;
}

```

[蓝桥杯青少年组国赛 2022] 最少问题

点 i 可以往 $i+1 \sim i+w[i]$ 连边, 然后从起点跑最短路

```

#include <bits/stdc++.h>

using namespace std;

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

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) {
        int x; cin >> x;

```

```

    for (int j = i+1; j<=i+x && j<=n; ++j) vec[i].push_back(j);
}

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

cout << dis[n] << endl;
return 0;
}

```

Simple path

从起点往后搜, 把搜到的结果存到数组中, 搜到终点时, 进行输出

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 2e5 + 5;
vector<int> vec[maxn];
bool st[maxn];
int w[maxn], len = 0;
int n, x, y;

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

    w[++len] = u;
    if (u == y) {
        for (int i = 1; i <= len; ++i) cout << w[i] << " ";
        cout << endl;
    }
    for (int i : vec[u]) dfs(i);
    --len;
}

int main()
{
    cin >> n >> x >> y;
    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(x);
return 0;
}
```

Takahashi Tour

对树做 dfs 遍历

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

using namespace std;

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

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

    cout << u << " ";
    for (int i : vec[u]) {
        if (!st[i]) {
            dfs(i);
            cout << u << " ";
        }
    }
}

int main()
{
    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);
    }

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

    dfs(1);
    cout << endl;
    return 0;
}
```

Collision

从根开始遍历, 求到每个点的最短路

按理说, 求最短路需要用 bfs, 但是这个题是个树, 所以求最短路可以用 dfs

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 2e5 + 5;
vector<int> vec[maxn];
bool st[maxn];
int dis[maxn];

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

    for (int i : vec[u]) {
        if (!st[i]) {
            dis[i] = dis[u]+1;
            dfs(i);
        }
    }
}

int main()
{
    int n, m; cin >> n >> m;
    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);

    while (m -- ) {
        int a, b; cin >> a >> b;
        int d = dis[a] + dis[b];
        if (d&1) cout << "Road" << endl;
        else cout << "Town" << endl;
    }
    return 0;
}

```

Cycle

先求 1 到其他点的最短路, 然后看谁能绕回 1, 求一个最小环

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 2e5 + 5;
const int inf = 0x3f3f3f3f;

```

```
vector<int> vec[maxn];
vector<int> vv;
int dis[maxn];

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

int main()
{
    int n, m; cin >> n >> m;
    while (m -- ) {
        int u, v; cin >> u >> v;
        vec[u].push_back(v);
        if (v == 1) vv.push_back(u);
    }

    bfs(n);

    int res = inf;
    for (int i : vv) {
        if (dis[i] != -1) res = min(res, dis[i]+1);
    }

    cout << (res==inf ? -1 : res) << endl;
    return 0;
}
```