

E5 E题

题目描述

给定一个有 n 个点、 m 条边的无向图，每条边的权值视为1，求从点 s 到点 t 的最短路数量，只要两条最短路的边不完全一样，就视为两个最短路。输出答案对 998244353取模后的结果。

题解思路

最自然的想法是改造Dijkstra算法等最短路算法，但是都很复杂。

观察这道题的特殊性发现，所有的边权都为1。在以 s 点为根节点的广度优先搜索树当中，如果 s 到 t 存在一条最短路，那么 t 节点之前一定有深度比他小的节点。所以 t 点到 s 点的最短路长度就是 t 点在以 s 点为根节点的广度优先搜索树当中的深度。用每个直接联通、深度比当前少一的点来更新路径。

代码

```
#include <bits/stdc++.h>
#define rep(i,a,b) for(int i=a;i<=b;i++)
typedef long long ll;
using namespace std;
const ll maxN = 1e6+5, maxM = 16e5+5, mod = 998244353, inf = LONG_MAX;
vector<int> graph[maxN];
ll dep[maxN];
bool v[maxN];
int cnt[maxN];
int main(){
    int m,n,s,t;
    scanf("%d%d%d%d",&n,&m,&s,&t);
    rep(i,1,m){
        int u,v;
        scanf("%d%d",&u,&v);
        graph[u].push_back(v);
        graph[v].push_back(u);
    }
    queue<int> q;
    dep[s] = 0;
    v[s] = 1;
    cnt[s] = 1;
    q.push(s);
    while(!q.empty()){
        int front=q.front();q.pop();
        for(int i=0; i<graph[front].size(); i++){
            int temp = graph[front][i];
            if(!v[temp]) {
                v[temp]=1;
                dep[temp] = dep[front]+1;
                q.push(temp);
            }
            if(dep[temp] == dep[front]+1) {
                cnt[temp] = (cnt[temp]+cnt[front])%mod;
            }
        }
    }
}
```

```
        }  
    }  
}  
printf("%d\n",cnt[t]);  
return 0;  
}
```