

E4F 魔法舞步

题目描述

n 点 m 边的有向图，每次可以移动三步，问是否可以从 S 走到 T 以及其最少移动次数

解题思路

状态设计为 $[i][0/1/2]$ 表示走到 i 步数 $step \equiv 0/1/2 \pmod{3}$ 的最短路径长度

即状态变为原来的三倍，直接按新的状态跑Dijkstra或者SPFA即可

代码实现

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

using namespace std;

using ll = long long;
using db = long double; // or double, if TL is tight
using str = string;
using u32 = unsigned int;
using u64 = unsigned long long;
using u128 = unsigned __int128;

using pii = pair<int, int>;
using pll = pair<ll, ll>;
using pdd = pair<db, db>;
#define fi first
#define se second

#define vc vector
using vi = vc<int>;
using vb = vc<bool>;
using vll = vc<ll>;
using vd = vc<db>;
using vs = vc<str>;
using vch = vc<char>;
using vpii = vc<pii>;
using vpll = vc<pll>;
using vpdd = vc<pdd>;
using vvi = vc<vc<int>>;
using vvll = vc<vc<ll>>;
using vvch = vc<vc<char>>;
using vvb = vc<vc<bool>>;
```

```

using vvp11 = vc<vc<p11>>;
using vvp11 = vc<vc<p11>>;
using vvvi = vc<vc<vc<int>>>>;

const int mod = 998244353; // 1e9 + 7;
const int INF = 0x3f3f3f3f;
const ll BIG = 1e18; // not too close to LLONG_MAX
const db PI = acos((db)-1);
const int dx[4]{1, 0, -1, 0}, dy[4]{0, 1, 0, -1}; // for
every grid problem!!
mt19937
rng32((uint32_t)chrono::steady_clock::now().time_since_epo
ch().count());
mt19937_64
rng64((uint64_t)chrono::steady_clock::now().time_since_epo
ch().count());

#define cl(x, y) memset(x, y, sizeof(x))
#define de(x) cerr << #x << " = " << x << " "
#define del(x) cerr << #x << " = " << x << endl

#define overload3(a, b, c, d, ...) d
#define rep1(a) for(int i = 0; i < (a); i++)
#define rep2(i, a) for (int i = 0; i < (a); i++)
#define rep3(i, a, b) for (int i = (a); i <= (b); i++)
#define per1(a) for(int i = (a) - 1; i >= 0; i--)
#define per2(i, a) for (int i = (a) - 1; i >= 0; i--)
#define per3(i, a, b) for (int i = (a); i >= (b); i--)
#define rep(args...) overload3(args, rep3, rep2, rep1)
(args)
#define per(args...) overload3(args, per3, per2, per1)
(args)
#define for_subset(t, s) \
    for (ll t = (s); t >= 0; t = (t == 0 ? -1 : (t - 1) &
(s)))

#define co cout
#define dl '\n'
#define ret return
#define pb push_back
#define tcT template <class T
#define YES cout << "YES" << endl
#define Yes cout << "Yes" << endl
#define NO cout << "NO" << endl
#define No cout << "No" << endl
#define INT(args...) \
    int args; \
    rd(args)
#define LL(args...) \
    ll args; \
    rd(args)

```

```

#define all(s) s.begin(), s.end()
#define rall(s) s.rbegin(), s.rend()
#define sz(v) int(v.size())
#define SUM(a) accumulate(all(a), 0ll)
#define MIN(v) (*min_element(all(v)))
#define MAX(v) (*max_element(all(v)))
#define unisort(v) \
    sort(all(v)); \
    v.resize(unique(all(v)) - v.begin())

int popcnt(int x) { return __builtin_popcount(x); }
int popcnt(u32 x) { return __builtin_popcount(x); }
int popcnt(ll x) { return __builtin_popcountll(x); }
int popcnt(u64 x) { return __builtin_popcountll(x); }

#define LOWBIT(x) ((x) & (-x))
// (0, 1, 2, 3, 4) -> (-1, 0, 1, 1, 2)
int topbit(int x) { return (x == 0 ? -1 : 31 -
__builtin_clz(x)); }
int topbit(u32 x) { return (x == 0 ? -1 : 31 -
__builtin_clz(x)); }
int topbit(ll x) { return (x == 0 ? -1 : 63 -
__builtin_clzll(x)); }
int topbit(u64 x) { return (x == 0 ? -1 : 63 -
__builtin_clzll(x)); }
// (0, 1, 2, 3, 4) -> (-1, 0, 1, 0, 2)
int lowbit(int x) { return (x == 0 ? -1 :
__builtin_ctz(x)); }
int lowbit(u32 x) { return (x == 0 ? -1 :
__builtin_ctz(x)); }
int lowbit(ll x) { return (x == 0 ? -1 :
__builtin_ctzll(x)); }
int lowbit(u64 x) { return (x == 0 ? -1 :
__builtin_ctzll(x)); }
tcT > T cdiv(T &a, T &b) {
    return a / b + ((a ^ b) > (T)(0) && a % b);
} // divide a by b rounded up
tcT > T fdiv(T &a, T &b) {
    return a / b - ((a ^ b) < (T)(0) && a % b);
} // divide a by b rounded down

tcT, class S > inline ostream &operator<<(ostream &i,
const pair<T, S> &a) { return i << a.first << ' ' <<
a.second; }
tcT, class S > inline istream &operator>>(istream &i,
pair<T, S> &a) { return i >> a.first >> a.second; }
tcT > inline ostream &operator<<(ostream &i, const
vector<T> &a) {
    if (a.size())
        i << a[0];
    for (int j = 1; j < int(a.size()); j++)
        i << ' ' << a[j];
}

```

```

        return i;
    }
    tcT > inline istream &operator>>(istream &i, vector<T> &a)
    {
        for (auto &j : a)
            i >> j;
        return i;
    }
    void err(istream_iterator<string> it) {
        it.~istream_iterator(); }
    template <typename T, typename... Args>
    void err(istream_iterator<string> it, T a, Args... args) {
        cerr << *it << " = " << a << endl;
        err(++it, args...);
    }
    void wt() { cout << '\n'; }
    tcT, class... Args > void wt(const T &a, const Args
    &...args) {
        cout << a;
        (void)(cout << ... << (cout << ' ', args));
        cout << '\n';
    }
    tcT > void wt(vector<vector<T>> &a) {
        for (auto &i : a)
            wt(i);
    }
    tcT, class S > void wt(vector<pair<T, S>> &a) {
        for (auto &i : a)
            wt(i);
    }
    template <class... T>
    void rd(T &...a) { (cin >> ... >> a); }

#define LOCAL_TEST freopen("out.txt", "w", stdout)
#define IOS
ios::sync_with_stdio(false);cin.tie(0);cout.tie(0)
#define FIX(x) cout << fixed << setprecision(x)

const int maxn = 1e5 + 5;

inline void INIT() {

}

struct node {
    int x, s;

    friend bool operator < (node a, node b) {
        return a.s < b.s;
    }
};

```

```

inline vvl1 dijkstra(int sta, int n, vvp11 edge) {
    vvl1 dis(n, v11(3, 0x3f3f3f3f3f3f3f3f));
    dis[sta][0] = 0;
    vvb jud(n, vb(3));
    priority_queue<pair<11, node>, vector<pair<11, node>>,
greater<pair<11, node>>> q;
    q.push(make_pair(dis[sta][0], (node){sta, 0}));
    while(!q.empty()) {
        pair<11, node> now = q.top();
        q.pop();
        if(jud[now.second.x][now.second.s]) continue;
        jud[now.second.x][now.second.s] = true;
        int nxt = (now.second.s + 1) % 3;
        for(auto &[ed, val]: edge[now.second.x]) {
            if(dis[ed][nxt] > now.first + val) {
                dis[ed][nxt] = now.first + val;
                q.push(make_pair(dis[ed][nxt], (node){ed,
nxt}));
            }
        }
    }
    return dis;
}

inline void SOLVE() {

    INT(n, m);
    vvp11 g(n);
    rep(m) {
        INT(u, v);
        u--, v--;
        g[u].pb({v, 1});
    }
    INT(s, t);
    s--, t--;

    vvl1 dis = dijkstra(s, n, g);

    if(dis[t][0] == 0x3f3f3f3f3f3f3f3f) wt(-1);
    else wt(dis[t][0] / 3);

}

/*

*/

int main() {

    IOS; FIX(20);

    INIT();

```

```
int tt = 1;
// cin >> tt;
while(tt--)
    SOLVE();

// cout << "program ends..." << endl;
// system("pause");
return 0;
}
```