

# 差分约束

## 人员

赵熙羽、陈洛冉、杨瑾硕、周子一、于子珈、秦显森 到课, 司云心、李子瀚 线上

## 上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/2330>

The screenshot shows a competition results page with a table of scores. The table has columns for #, 用户名 (Username), 姓名 (Name), 编程分 (Programming Score), 时间 (Time), A, B, C, D, and E. The table data is as follows:

#	用户名	姓名	编程分	时间	A	B	C	D	E
1	yuzijia1	于子珈	500	2036	100	100	100	100	100
2	zhaoxiyu	赵熙羽	400	1781	100	100	100		100
3	niutongze	牛同泽	400	1806	100	100	100		100
4	qinxiansen	秦显森	300	1612	100	100	100		
5	xieyakai	谢亚锴	300	1746	100	100	100		
6	chenluoran	陈洛冉	283	617	100	100	83		
7	lizihan	李子瀚	200	454	100	100			
8	siyunxin	司云心	200	476	100	100			
9	yangjinshuo	杨瑾硕	200	610	100	100			
10	zhouziyi	周子一	200	666	100	100			

At the bottom left, there is a button labeled "10 点点什么...". At the bottom right, a message says "您正在共享屏幕" and a red button says "结束共享".

## 本周作业

<https://cppoj.kids123code.com/contest/2454> (课上讲了 A ~ C 题, 课后作业是 C 题)

## 课堂表现

今天课上讲了 差分约束 这个知识点, 反映出来同学们对 SPFA 这个算法不熟的问题, 同学们课下要多复习复习前面学的内容。

## 课堂内容

### [USACO07FEB] Cow Party S (上周作业)

用原图跑一遍 dijkstra, 用反图跑一遍 dijkstra 即可

```
#include <bits/stdc++.h>
#define int long long

using namespace std;

const int maxn = 2e6 + 5;
const int inf = 0x3f3f3f3f3f3f3f3f;
struct eInfo {
    int u, v, w;
    eInfo() {}
    eInfo(int u, int v, int w) : u(u), v(v), w(w) {}
}
```

```

    int to, value;
};

vector<eInfo> vec[maxn];

struct node {
    int id, d;
    bool operator < (const node& p) const { return d < p.d; }
    bool operator > (const node& p) const { return d > p.d; }
};

int dis[maxn];
bool st[maxn];

void dijkstra(int _st) {
    priority_queue<node, vector<node>, greater<node>> q;
    q.push({_st, 0}); dis[_st] = 0;
    while (!q.empty()) {
        node u = q.top(); q.pop();
        int id = u.id, d = u.d;
        if (st[id]) continue;
        st[id] = true;

        for (eInfo it : vec[id]) {
            if (dis[it.to] > d+it.value) {
                dis[it.to] = d+it.value; q.push({it.to, dis[it.to]});
            }
        }
    }
}

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

    memset(dis, 0x3f, sizeof(dis)), memset(st, false, sizeof(st));
    dijkstra(id), dijkstra(n+id);
    int res = 0;
    for (int i = 1; i <= n; ++i) res = max(res, dis[i] + dis[n+i]);
    cout << res << endl;
    return 0;
}

```

## 【模板】差分约束

把  $x_i - x_j \leq c$

转化为  $x_i \leq x_j + c$

转化为一条点 j 到点 i 的, 边权为 c 的边即可

设定一个超级源点, 从超级源点往所有点连边, 跑一遍最短路即可, 因为有负边, 所以需要用 SPFA 求最短路

如果图中有负环, 说明无解

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

using namespace std;

const int maxn = 5e3 + 5;
const int inf = 0x3f3f3f3f;
struct node {
    int to, value;
};
vector<node> vec[maxn];
int dis[maxn], cnt[maxn];
bool st[maxn];

int main()
{
    int n, m; cin >> n >> m;
    while (m -- ) {
        int a, b, c; cin >> a >> b >> c;
        vec[b].push_back({a,c});
    }
    for (int i = 1; i <= n; ++i) vec[0].push_back({i,0});

    memset(dis, 0x3f, sizeof(dis));
    queue<int> q; q.push(0); dis[0] = 0; st[0] = true;
    while (!q.empty()) {
        int u = q.front(); q.pop();
        st[u] = false;
        for (node it : vec[u]) {
            if (dis[u]+it.value < dis[it.to]) {
                dis[it.to] = dis[u]+it.value; ++cnt[it.to];
                if (cnt[it.to] == n) { cout << "NO" << endl; return 0; }
                if (!st[it.to]) { q.push(it.to); st[it.to] = true; }
            }
        }
    }

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

## Hopscotch Addict

分层图, 把 1 个点拆成 3 个点, 然后从起点往终点跑 bfs 求最短路

```

#include <bits/stdc++.h>

using namespace std;

int get_id(int a, int b) { return a*3+b; }

const int maxn = 3e5 + 100;
vector<int> vec[maxn];
int dis[maxn];

int bfs(int a, int b) {
    int _st = get_id(a, 0), _ed = get_id(b, 0);
    memset(dis, -1, sizeof(dis));

    queue<int> q; q.push(_st); dis[_st] = 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;
        }
    }

    return (dis[_ed]==-1 ? dis[_ed] : dis[_ed]/3);
}

int main()
{
    int n, m; cin >> n >> m;
    while (m -- ) {
        int a, b; cin >> a >> b;
        for (int i = 0; i < 3; ++i) {
            int a_id = get_id(a, i), b_id = get_id(b, (i+1)%3);
            vec[a_id].push_back(b_id);
        }
    }

    int a, b; cin >> a >> b;
    cout << bfs(a, b) << endl;
    return 0;
}

```

## 小 K 的农场

按照题目的要求建边, 可以把原问题转化为一个差分约束问题

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 5e3 + 5;

```

```
const int inf = 0x3f3f3f3f;
struct node {
    int to, value;
};
vector<node> vec[maxn];
int dis[maxn], cnt[maxn];
bool st[maxn];

int main()
{
    int n, m; cin >> n >> m;
    while (m -- ) {
        int op; cin >> op;
        if (op == 1) {
            int a, b, c; cin >> a >> b >> c;
            vec[a].push_back({b,-c});
        } else if (op == 2) {
            int a, b, c; cin >> a >> b >> c;
            vec[b].push_back({a,c});
        } else {
            int a, b; cin >> a >> b;
            vec[a].push_back({b,0}), vec[b].push_back({a,0});
        }
    }
    for (int i = 1; i <= n; ++i) vec[0].push_back({i,0});

    memset(dis, 0x3f, sizeof(dis));
    queue<int> q; q.push(0); dis[0] = 0; st[0] = true;
    while (!q.empty()) {
        int u = q.front(); q.pop();
        st[u] = false;
        for (node it : vec[u]) {
            if (dis[u]+it.value < dis[it.to]) {
                dis[it.to] = dis[u]+it.value; ++cnt[it.to];
                if (cnt[it.to] == n) { cout << "No" << endl; return 0; }
                if (!st[it.to]) { q.push(it.to); st[it.to] = true; }
            }
        }
    }

    cout << "Yes" << endl;
    return 0;
}
```