

## #E4题解

## 题目分析

## G 旅行计划

时间限制: 1500ms 内存限制: 65536kb

通过率: 18/28 (64.29%) 正确率: 18/240 (7.50%)

## 题目描述

教练们很累，写不动题目背景力！

形式化地说，给定一张  $n$  个点  $m$  条边的带权无向连通图。你需要处理  $q$  次询问，第  $i$  次询问的内容为从点  $s_i$  到点  $t_i$  经过不超过  $k_i$  条边的最短路径长度是多少。

## 输入格式

第一行三个整数  $n, m, q$  ( $1 \leq n \leq 100, n-1 \leq m \leq \frac{n(n-1)}{2}, 1 \leq q \leq 10^6$ )，含义同题目描述。接下来  $m$  行，每行三个正整数  $u_i, v_i, w_i$  ( $1 \leq u_i < v_i \leq n, 1 \leq w_i \leq 10^9$ )，表示第  $i$  条边  $(u_i, v_i)$  的权值为  $w_i$ 。接下来  $q$  行，每行三个正整数  $s_i, t_i, k_i$  ( $1 \leq s_i, t_i \leq n, 1 \leq k_i \leq m$ )，含义同题目描述。

保证图连通且没有重边和自环。

## 输出格式

输出  $q$  行，第  $i$  ( $1 \leq i \leq q$ ) 行一个整数：

- 如果不存在一条从  $s_i$  到  $t_i$  且经过边数不超过  $k_i$  的路径，输出  $-1$ 。
- 否则输出从点  $s_i$  到点  $t_i$  经过不超过  $k_i$  条边的最短路径长度。

可以设  $f[k][s][t]$ ，表示从点  $s$  到点  $t$  不超过  $k$  条边的最短路径

$n$  比较小， $q$  比较大  
由此可知，我们要先求出所有的  $f[k][s][t]$   
再进行输出

可知  $f[k][s][t]$  的最大值为  $n \cdot w = 10^{11}$ ，需要用 long long

- 由于  $f[k][s][t]$  之间有很强的关联，想到可以用三维dp
- 由于最短路径不可能有环路，所以  $k \leq n$

得到状态转移方程：

$$f[t+1][i][j] = \min(f[t][i][k] + f[1][k][j]); \quad 1 \leq k \leq n$$

表示从  $k$  点一步到  $j$ 特别的，当  $k=j$  时， $f[1][j][j]=0$ ，表示在  $j$  点没有移动

```
for (int t = 1; t < n; t++)
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= n; j++)
        {
            f[t+1][i][j] = f[t][i][j];
            for (int k = 1; k <= n; k++)
                f[t+1][i][j] = min(f[t+1][i][j], f[t][i][k] + f[1][k][j]);
        }
```

前期准备:

#define oo 1e18表示没有接通

将f[k][i][j]都初始化为oo

将f[k][i][i]赋值为0, 表示没有动

输入f[1][u][v]表示可以一步从u到v

输出:

当k>n时, 将其变为n

f[k][s][t]==oo时, 代表没有通路, 输出-1

```
for (int i = 0; i <= n; i++)
    for (int j = 1; j <= n; j++)
        for (int k = 1; k <= n; k++)
            f[i][j][k] = oo;
for (int x = 0; x <= n; x++)
    for (int i = 1; i <= n; i++)
        f[x][i][i] = 0;
for (int i = 1; i <= m; i++)
{
    int u, v, w;
    scanf("%d%d%d", &u, &v, &w);
    f[1][u][v] = min(f[1][u][v], 1ll * w);
    f[1][v][u] = min(f[1][v][u], 1ll * w);
}
```

```
while (q--)
{
    int s, t, k;
    scanf("%d%d%d", &s, &t, &k);
    k = min(k, n);
    long long ans = f[k][s][t];
    if (ans == oo)
        ans = -1;
    printf("%lld\n", ans);
}
```

##代码:

```
#include <cstdio>
#include <algorithm>
#include <queue>
#include <cassert>

using namespace std;

const int N = 105;
const long long oo = 1e18;

int n, m, q;
long long f[N][N][N];

int main()
{
    scanf("%d%d%d", &n, &m, &q);
    for (int i = 0; i <= n; i++)
        for (int j = 1; j <= n; j++)
            for (int k = 1; k <= n; k++)
                f[i][j][k] = oo;
    for (int x = 0; x <= n; x++)
        for (int i = 1; i <= n; i++)
            f[x][i][i] = 0;
    for (int i = 1; i <= m; i++)
    {
        int u, v, w;
        scanf("%d%d%d", &u, &v, &w);
        f[1][u][v] = min(f[1][u][v], 1ll * w);
        f[1][v][u] = min(f[1][v][u], 1ll * w);
    }
    for (int t = 1; t < n; t++)
```

```
        for (int i = 1; i <= n; i++)
            for (int j = 1; j <= n; j++)
            {
                f[t + 1][i][j] = f[t][i][j];
                for (int k = 1; k <= n; k++)
                    f[t + 1][i][j] = min(f[t + 1][i][j], f[t][i][k] + f[1][k][j]);
            }
    while (q--)
    {
        int s, t, k;
        scanf("%d%d%d", &s, &t, &k);
        k = min(k, n);
        long long ans = f[k][s][t];
        if (ans == oo)
            ans = -1;
        printf("%lld\n", ans);
    }
    return 0;
}
```