

Telephone Lines

一. 考察内容:

图论 最短路 二分答案

二. 题目分析:

[写题思路]

这是一道解法很巧妙的题，我们二分一个答案，对于该答案，把权值大于答案的边的权值设为1，其余边权值设为0，如果最短路 $\leq k$ ，则该答案可行，否则该答案不可行，最后输出二分的答案即可。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATE
/*****
*创建时间: 2018 08 27
*文件类型: 源代码文件
*题目来源: POJ
*当前状态: 已通过
*备忘录: 图论 最短路 二分答案
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
// #include <sys/wait.h>
// #include <sys/types.h>
// #include <unistd.h>
using namespace std;
#define MAXN 1001
#define MAXM 10001
int n,m,k,head[MAXN],_edge,d[MAXN],come[MAXM];
struct EDGE
{
    int a,b,v,next;
    EDGE(int a=0,int b=0,int v=0,int next=0):a(a),b(b),v(v),next(next){}
}edge[2*MAXM],e[MAXM];
struct A
{
    int id,v;
    A(int id=0,int v=0):id(id),v(v){}
    bool operator < (A x) const
    {
        return v>x.v;
    }
};
/* Variable explain:
*/
void adde(int a,int b,int v)
{
    // printf("%d %d %d\n",a,b,v);
    edge[++_edge]=EDGE(a,b,v,head[a]);
    head[a]=_edge;
}
void reset()
{
    for(int i=1;i<=n;++i)head[i]=0,d[i]=1e9;
    _edge=d[1]=0;
}
```

```

void read()
{
    // freopen("phoneline.in","r",stdin);
    // freopen("phoneline.out","w",stdout);
    scanf("%d%d%d",&n,&m,&k);
    for(int i=1;i<=m;++i) scanf("%d%d%d",&e[i].a,&e[i].b,&e[i].v);
    return;
}
void dijkstra()
{
    bool vis[MAXN]={0};
    priority_queue <A> q;
    for(int i=1;i<=n;++i) q.push(A(i,d[i]));
    while(!q.empty())
    {
        A a=q.top();
        q.pop();
        if(vis[a.id]) continue;
        vis[a.id]=1;
        // printf("#%d#\n",a.id);
        for(int i=head[a.id];i;i=edge[i].next)
        {
            int b=edge[i].b,v=edge[i].v;
            // printf("%d\n",b);
            if(d[b]>d[a.id]+v)
            {
                d[b]=d[a.id]+v;
                come[b]=a.id;
                q.push(A(b,d[b]));
            }
        }
    }
}
bool judge(int x)
{
    reset();
    for(int i=1;i<=m;++i)
        if(e[i].v>x) adde(e[i].a,e[i].b,1),adde(e[i].b,e[i].a,1);
        else adde(e[i].a,e[i].b,0),adde(e[i].b,e[i].a,0);
    dijkstra();
    // for(int i=1;i<=n;++i) printf("%d ",d[i]);
    return d[n]<=k;
}
void solve()
{
    int l=0,r=1000000,ans=-1;
    while(l<=r)
    {
        int m=(l+r)>>1;
        if(judge(m)) r=m-1,ans=m;
        else l=m+1;
    }
    printf("%d\n",ans);
}
int main()
{
    read();
    // printf("%d\n",judge(4));
    solve();
    return 0;
}

```

[<题目跳转>](#) [<查看代码>](#)