

[报告]CODEFROCES Volleyball

[Source]

<http://codeforces.com/problemset/problem/96/D>

[Description]

某人要回家，然后给了一个图，每个点上有可以多次使用的出租车，估计出租车的油有限制，每个地点的车的行驶距离有限制，而且车不能停在半路，每个车还有花费，然后某人要省钱，问最小话费，无解为-1。

[Solution]

由于是稀疏图（与边同一等级），所以先在原图上 N 次 spfa，求出任意两点的最短距，然后根据每个点的出租车的行驶范围与该点到其他点得距离重新构图，权为出租车的花费，再来一次 spfa。

[Code]

```
#include<iostream>
#include<cstdio>
#include<string.h>
#include<string>
#include<math.h>
#include<algorithm>
#include<queue>
using namespace std;
#define maxn 1009
#define inf 1LL<<60

struct edge{
    int v;
    int next;
    int w;
};

edge e[maxn*maxn];
int m,n,x,y,nn;
long long dis[maxn][maxn];
bool flag[maxn];
int head[maxn];
```

```

void addedge(int u,int v,int w){
    e[nn].v=v;
    e[nn].w=w;
    e[nn].next=head[u];
    head[u]=nn++;
}

void spfa(int start){
    queue<int> que;
    for(int i=1;i<=n;i++) dis[start][i]=inf;
    memset(flag,false,sizeof(flag));
    que.push(start);
    dis[start][start]=0;
    while(!que.empty()){
        int x=que.front();
        que.pop();
        flag[x]=false;
        for(int i=head[x];i!=-1;i=e[i].next){
            if(dis[start][e[i].v]>dis[start][x]+e[i].w){
                dis[start][e[i].v]=dis[start][x]+e[i].w;
                if(!flag[e[i].v]){
                    flag[e[i].v]=1;
                    que.push(e[i].v);
                }
            }
        }
    }
}

int main(){
    scanf("%d %d",&n,&m);
    scanf("%d %d",&x,&y);
    nn=0;
    memset(head,-1,sizeof(head));
    for(int i=0;i<m;i++){
        int u,v,w;
        scanf("%d %d %d",&u,&v,&w);
        addedge(u,v,w);
        addedge(v,u,w);
    }
    for(int i=1;i<=n;i++)
        spfa(i);
    nn=0;
}

```

```

memset(head,-1,sizeof(head));
for(int i=1;i<=n;i++){
    int u,v;
    scanf("%d %d",&u,&v);
    for(int j=1;j<=n;j++){
        if(i!=j&&dis[i][j]<=u)
            addedge(i,j,v);
    }
}
spfa(x);
if(dis[x][y]==inf) cout << "-1" << endl;
else cout << dis[x][y] << endl;
return 0;
}

```