**Choose the best route**

**Time Limit: 2000/1000 MS (Java/Others)    Memory Limit: 32768/32768 K (Java/Others)  
Total Submission(s): 12423    Accepted Submission(s): 4039**

Problem Description

One day , Kiki wants to visit one of her friends. As she is liable to carsickness , she wants to arrive at her friend’s home as soon as possible . Now give you a map of the city’s traffic route, and the stations which are near Kiki’s home so that she can take. You may suppose Kiki can change the bus at any station. Please find out the least time Kiki needs to spend. To make it easy, if the city have n bus stations ,the stations will been expressed as an integer 1,2,3…n.

Input

There are several test cases.   
Each case begins with three integers n, m and s,(n<1000,m<20000,1=<s<=n) n stands for the number of bus stations in this city and m stands for the number of directed ways between bus stations .(Maybe there are several ways between two bus stations .) s stands for the bus station that near Kiki’s friend’s home.  
Then follow m lines ,each line contains three integers p , q , t (0<t<=1000). means from station p to station q there is a way and it will costs t minutes .  
Then a line with an integer w(0<w<n), means the number of stations Kiki can take at the beginning. Then follows w integers stands for these stations.

Output

The output contains one line for each data set : the least time Kiki needs to spend ,if it’s impossible to find such a route ,just output “-1”.

Sample Input

5 8 5

1 2 2

1 5 3

1 3 4

2 4 7

2 5 6

2 3 5

3 5 1

4 5 1

2

2 3

4 3 4

1 2 3

1 3 4

2 3 2

1

1

Sample Output

1

-1

Author

dandelion

Source

[2009浙江大学计算机研考复试（机试部分）——全真模拟](http://acm.hdu.edu.cn/search.php?field=problem&key=2009%D5%E3%BD%AD%B4%F3%D1%A7%BC%C6%CB%E3%BB%FA%D1%D0%BF%BC%B8%B4%CA%D4%A3%A8%BB%FA%CA%D4%B2%BF%B7%D6%A3%A9%A1%AA%A1%AA%C8%AB%D5%E6%C4%A3%C4%E2&source=1&searchmode=source)

分析：

把起点看成0的位置 把Kiki可以开始的点与起点的权值赋0 然后再找到s的最短路就可以了

注意：

从p到q是单向的 （被这个地方坑到了）

构造有向图 完美ac

AC代码：

#include <iostream>

#include <cstdio>

#include <cstring>

#define INF 0x3f3f3f3f

#define MAX 2000

using namespace std;

int vis[MAX],dis[MAX],map[MAX][MAX];

int n,m,s;

void dij()

{

memset(dis,0,sizeof(dis));

for(int i = 0; i <= n; i++)

vis[i] = map[0][i];

dis[0] = 1;

int min,k;

for(int i = 0; i <= n; i++)

{

min = INF;

for(int j = 0; j <= n; j++)

{

if(!dis[j]&&vis[j] < min)

{

min = vis[j];

k = j;

}

}

if(min == INF)

break;

dis[k] = 1;

for(int j = 0; j <= n; j++)

{

if(dis[j]==0&&vis[j] > vis[k] + map[k][j])

{

vis[j] = vis[k] + map[k][j];

}

}

}

}

int main()

{

int p,q,t;

while(~scanf("%d%d%d",&n,&m,&s))

{

for(int i = 0; i <= n; i++)

for(int j = 0; j <= n; j++)

map[i][j] = INF;

for(int i = 1; i <= m; i++)

{

scanf("%d%d%d",&p,&q,&t);

if(t < map[p][q])

map[p][q] = t;

}

int w,h;

scanf("%d",&w);

for(int i = 1; i <= w; i++)

{

scanf("%d",&h);

map[0][h] = 0;

}

/\*for(int i = 0; i <= n; i++)

{

for(int j = 0; j <= n; j++)

printf("%d ",map[i][j]);

printf("\n");

}\*/

dij();

if(vis[s]!=INF)

printf("%d\n",vis[s]);

else

printf("-1\n");

}

return 0;

}