**HDU Today**

**Time Limit: 15000/5000 MS (Java/Others)    Memory Limit: 32768/32768 K (Java/Others)  
Total Submission(s): 23336    Accepted Submission(s): 5597**

Problem Description

经过锦囊相助，海东集团终于度过了危机，从此，HDU的发展就一直顺风顺水，到了2050年，集团已经相当规模了，据说进入了钱江肉丝经济开发区500强。这时候，XHD夫妇也退居了二线，并在风景秀美的诸暨市浬浦镇陶姚村买了个房子，开始安度晚年了。  
这样住了一段时间，徐总对当地的交通还是不太了解。有时很郁闷，想去一个地方又不知道应该乘什么公交车，在什么地方转车，在什么地方下车（其实徐总自己有车，却一定要与民同乐，这就是徐总的性格）。  
徐总经常会问蹩脚的英文问路：“Can you help me?”。看着他那迷茫而又无助的眼神，热心的你能帮帮他吗？  
请帮助他用最短的时间到达目的地（假设每一路公交车都只在起点站和终点站停，而且随时都会开）。

Input

输入数据有多组，每组的第一行是公交车的总数N(0<=N<=10000)；  
第二行有徐总的所在地start，他的目的地end；  
接着有n行，每行有站名s，站名e，以及从s到e的时间整数t(0<t<100)(每个地名是一个长度不超过30的字符串)。  
note：一组数据中地名数不会超过150个。  
如果N==-1，表示输入结束。

Output

如果徐总能到达目的地，输出最短的时间；否则，输出“-1”。

Sample Input

6

xiasha westlake

xiasha station 60

xiasha ShoppingCenterofHangZhou 30

station westlake 20

ShoppingCenterofHangZhou supermarket 10

xiasha supermarket 50

supermarket westlake 10

-1

Sample Output

50

Hint:

The best route is:

xiasha->ShoppingCenterofHangZhou->supermarket->westlake

虽然偶尔会迷路，但是因为有了你的帮助

\*\*和\*\*从此还是过上了幸福的生活。

――全剧终――

Author

lgx

Source

[ACM程序设计\_期末考试（时间已定！！）](http://acm.hdu.edu.cn/search.php?field=problem&key=ACM%B3%CC%D0%F2%C9%E8%BC%C6_%C6%DA%C4%A9%BF%BC%CA%D4%A3%A8%CA%B1%BC%E4%D2%D1%B6%A8%A3%A1%A3%A1%A3%A9&source=1&searchmode=source)

分析：求最短了 直接用Dijkstra就可以了 但是首先要处理好 字符串 应该怎么在数组里表示

这里就可以看出来map容器的强大之处了

容器可以定义为 map<string,int>s或者map<string,string>s或者map<int,string>s说的简单点就是 用第一个类型来作为key值 第二个类型作为value值 通过key值来寻找value值 key值不可变 value值可以被更新

这样的话 我们就可以把起点和终点定为 s[a] = 1,s[b] = 2; a是起点，b是终点

就这样 依次把字符串表示为s数组的下标 s数组的值代表共有多少个不同的点

然后在构建一个图 直接求短路就可以了

AC代码：

#include <iostream>

#include <cstdio>

#include <cstring>

#include <map>

#define INF 0x3f3f3f3f

#define MAX 200

using namespace std;

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

char a[32], b[32];

int c,ans,k,n;

void dij()

{

memset(s,0,sizeof(s));

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

vis[i] = dis[1][i];

s[1] = 1;

vis[1] = 0;

int min,k;

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

{

min = INF;

for(int j = 1; j <= ans; j++)

{

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

{

min = vis[j];

k = j;

}

}

if(min == INF)

break;

s[k] = 1;

for(int j = 1; j <= ans; j++)

{

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

{

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

}

}

}

}

int main()

{

map<string,int>s;

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

{

if(n==-1)

break;

k = 0, ans = 3;

s.clear();

scanf("%s%s",a,b);

if(!strcmp(a,b))

k = 1;

s[a] = 1, s[b] = 2;

for(int i = 1; i <= 150; i++)

{

for(int j = 1; j <= 150; j++)

dis[i][j] = INF;

dis[i][i] = 0;

}

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

{

scanf("%s%s%d",a,b,&c);

if(!s[a]) s[a] = ans++;

if(!s[b]) s[b] = ans++;

if(c < dis[s[a]][s[b]])

dis[s[a]][s[b]] = dis[s[b]][s[a]] = c;

}

if(k == 1)

printf("0\n");

else

{

dij();

if(vis[2]==INF)

printf("-1\n");

else

printf("%d\n",vis[2]);

}

}

return 0;

}