**数据结构**

**——实验报告书**



姓名：樊雨晨

学号：201632110111

班级：软件工程(中外)16 班

学院：数理与信息工程学院

日期：2017年05月10日

# **实验十五 单源最短路Dijkstra算法**

## **问题**

## 某省自从实行了很多年的畅通工程计划后，终于修建了很多路。不过路多了也不好，每次要从一个城镇到另一个城镇时，都有许多种道路方案可以选择，而某些方案要比另一些方案行走的距离要短很多。这让行人很困扰。 现在，已知起点和终点，请你计算出要从起点到终点，最短需要行走多少距离。

## **实验思路**

贪心算法，从一个顶点出发，得到到各个点的距离，保存在一个二维数组中，然后找到距离最短的一个路径，将这个顶点和原有的顶点合为一个整体。修正到其他点的最短路径。

## **实验步骤**

一个二维数据，两个一维数组，二维数组储存各个顶点之间的权值，dist储存某一个顶点到其他定点的距离，s数组为标记数组，显示当前定点是否是属于已经在最短路径的集合中。

## **代码**

#**include**<iostream>

#**include**<algorithm>

#**include**<math.h>

#**define** maxsum 0x7FFFFFFF

**using** **namespace** std;

**int** map[1000][1000],dist[1000],s[1000];

**int** **Dijkstra**(**int** n,**int** S,**int** T){

**int** mindis,u;

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

dist[i]= map[S][i];

s[i]=0;

}

s[S]= 1;

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

mindis=maxsum; u=-1;

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

**if**(!s[j]&&dist[j]<mindis){

u=j; mindis = dist[j];

}

}

s[u]=1;

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

**if**(u==-1)

**break**;

**else**

**if**(!s[j] && dist[u]+map[u][j]<dist[j]&&map[u][j]<maxsum)

dist[j]=dist[u]+map[u][j];

}

printf("%d\n",dist[T]==maxsum?-1:dist[T]);

}

**int** **main**(){

**int** N,M;

**int** A,B,X;

**int** S,T;

**while**(~scanf("%d%d",&N,&M)){

**for**(**int** i=0;i<N;i++)

**for**(**int** j=0;j<N;j++)

map[i][j]=map[j][i]=maxsum;

**for**(**int** i=0;i<M;i++){

scanf("%d%d%d",&A,&B,&X);

**if**(map[A][B]>X)

map[A][B]=map[B][A]=X;

}

scanf("%d%d",&S,&T);

Dijkstra(N,S,T);

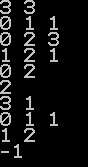
}

}

## **测试**

正常

## **结果**



## **总结**

特别要注意，在输入数据的时候可能存在重边的情况，所以需要将长边用短边替换掉，增加条件**if**(map[A][B]>X）。另外要定义一个值，表示两个顶点之间是不连通的。