**《离散数学》课程应用案例实践报告**

**班 级\_\_\_\_\_\_\_\_\_物联网1802\_\_\_\_\_\_\_\_\_\_**

**学 号\_\_\_\_\_\_\_\_\_8213180228\_\_\_\_\_\_\_\_\_\_**

**姓 名\_\_\_\_\_\_\_\_\_\_\_王云鹏\_\_\_\_\_\_\_\_\_\_\_\_**

**时 间\_\_\_\_\_\_\_2020年1月9日\_\_\_\_\_\_\_**

目录

[二、 问题背景 3](#_Toc29504142)

[三、 分析建模 3](#_Toc29504143)

[四、 问题描述 3](#_Toc29504144)

[五、 相关数学知识 4](#_Toc29504145)

[1. 无向图 4](#_Toc29504146)

[2. 赋权图 4](#_Toc29504147)

[3. 邻接矩阵 4](#_Toc29504148)

[六、 算法设计 5](#_Toc29504149)

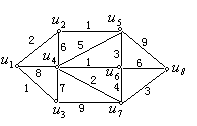
[1. Dijkstra算法介绍 5](#_Toc29504150)

[2. Dijkstra算法具体步骤 5](#_Toc29504151)

[七、 实验结果 6](#_Toc29504152)

# 问题背景

现有8个城市，它们之间的距离如下所示，其中小圆圈代表城市，连接线的长度代表它们之间的距离，现在一个u1城市的旅客想到其他任意一个城市去，请给出他到达其他城市的最优方案：



# 分析建模

题目要求最优选择，如果以路程为衡量指标，即求得该旅客到达其他城市的最短距离的路径。用图论的观点分析，即求节点u1到其余顶点的最短路径即可满足要求。

# 问题描述

将改图上的节点与距离关系转化到矩阵进行存储表示，以求清楚明了的表示需要的数据。因此写出该图的带权邻接矩阵如下：



因题目所给图（设为G）是无向图，故W是对称阵．

# 相关数学知识

## 无向图

G=<V,E>,其中：

①V是非空集合，称为 顶点集。

②E是V中元素构成的无序二元组的集合，称为 边集。

## 赋权图

若将图G的每一条边e都对应一个实数w(e)，称w(e)为边的权，并称图G为赋权图.

## 邻接矩阵

是表示顶点之间相邻关系的矩阵。设G=(V,E)是一个图，其中V={v1,v2,…,vn}。G的邻接矩阵是一个具有下列性质的n阶方阵：

①对无向图而言，邻接矩阵一定是对称的，而且对角线一定为零（在此仅讨论无向简单图），有向图则不一定如此。

②在无向图中，任一顶点i的度为第i列所有元素的和，在有向图中顶点i的出度为第i行所有元素的和，而入度为第i列所有元素的和。

③用邻接矩阵法表示图共需要n^2个空间，由于无向图的邻接矩阵一定具有对称关系，所以扣除对角线为零外，仅需要存储上三角形或下三角形的数据即可，因此仅需要n（n-1）/2个空间。

# 算法设计

## Dijkstra算法介绍

求G中从顶点u0到其余顶点的最短路

设G为赋权有向图或无向图，G边上的权均非负，对每个顶点，定义两个标记（，），其中:

：表从顶点u0到*v*的一条路的权．

：*v*的父亲点，用以确定最短路的路线

算法的过程就是在每一步改进这两个标记，使最终为从顶点u0到v的最短路的权．

## Dijkstra算法具体步骤

S：具有永久标号的顶点集

输入: G的带权邻接矩阵

（１）赋初值：令 S＝{}, =0

,令=,=,

（2）更新、： ,若>

则令=,= 

（3）设是使取最小值的中的顶点，

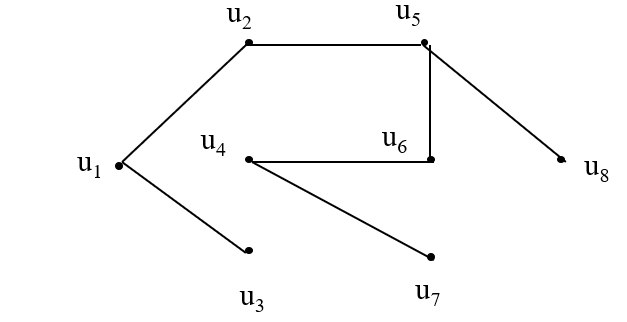
则令S=S∪{}，

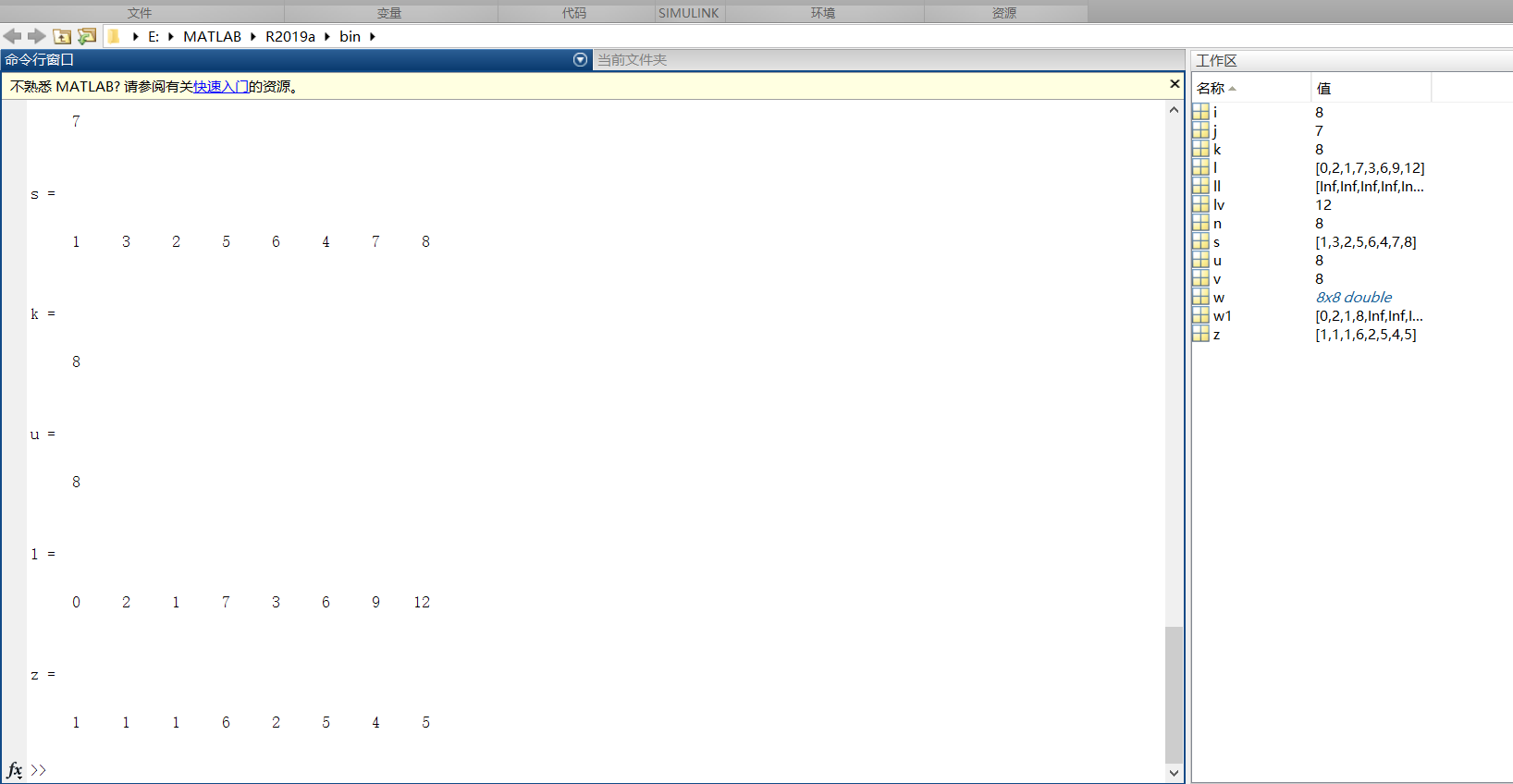
（4） 若φ，转2，否则，停止.

用上述算法求出的就是到的最短路的权，从的父亲标记追溯到, 就得到到的最短路的路线.

# 实验结果







【附录：代码】

%Matlab

clear all

close all

clc

w = [0 2 1 8 inf inf inf inf;

2 0 inf 6 1 inf inf inf;

1 inf 0 7 inf inf 9 inf;

8 6 7 0 5 1 2 inf;

inf 1 inf 5 0 3 inf 9;

inf inf inf 1 3 0 4 6;

inf inf 9 2 inf 4 0 3;

inf inf inf inf 9 6 3 0]

n=size(w,1);

w1=w(1,:);

%¸³³õÖµ

for i=1:n

l(i)=w1(i);

z(i)=1;

end

s=[];

s(1)=1;

u=s(1);

k=1

while k<n

% ¸üÐÂ l(v) ºÍ z(v)

for i=1:n

for j=1:k

if i~=s(j)

if l(i)>l(u)+w(u,i)

l(i)=l(u)+w(u,i);

z(i)=u;

end

end

end

end

%Çóv\*

ll = l;

for i=1:n

for j=1:k

if i~=s(j)

ll(i)=ll(i);

else

ll(i)=inf;

end

end

end

lv=inf;

for i=1:n

if ll(i)<lv

lv=ll(i);

v=i;

end

end

s(k+1)=v

k=k+1

u=s(k)

end