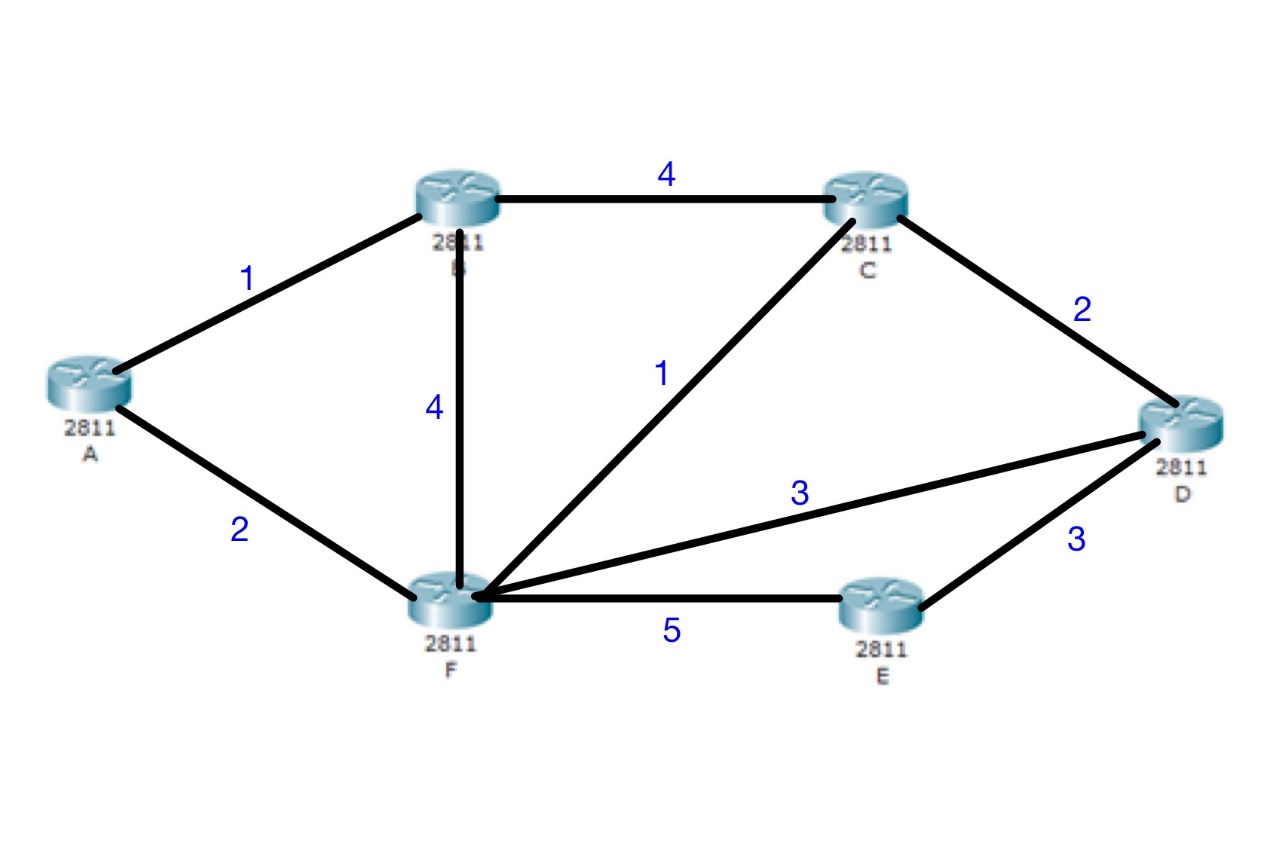
**问题描述：**

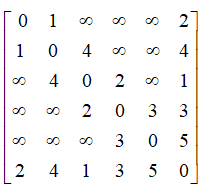
鉴于我们小组成员都来自计通学院，我们对专业知识计算机网络内的**路由器**进行研究。我们知道在整个互联网中，有着很多个小的无法互相连通的小网络，早在上世纪六十年代，针对不同网络无法互联的问题，路由器诞生了。路由器就是连通不同网络的中间媒介，在整个网络里又有无数个路由器相互连接通往不同的网络，使得网络内的主机能够相互通信，才有了今天的网络互连的时代。不难理解，在网络中，一台路由器传送信息到另一台路由器，为了节省开销和延时，必然要选择最短路径。在路由器中存在着路由表，记录着每个路由器与其下一跳路由器的距离（所有可达距离中的最短距离）。在计算机网络中，最常见的路由选择协议OSPF便解决了这个问题。接下来，我们将用数学实验所学知识，利用MATLAB实现路由器转发功能（OSPF协议的最短路径优先）的实现，即**路由算法**。

假设在某一网络中，存在着六台路由器，我们给这些路由器编号A,B，C，D，E，F。他们的在网络里的链路关系如下图所示：

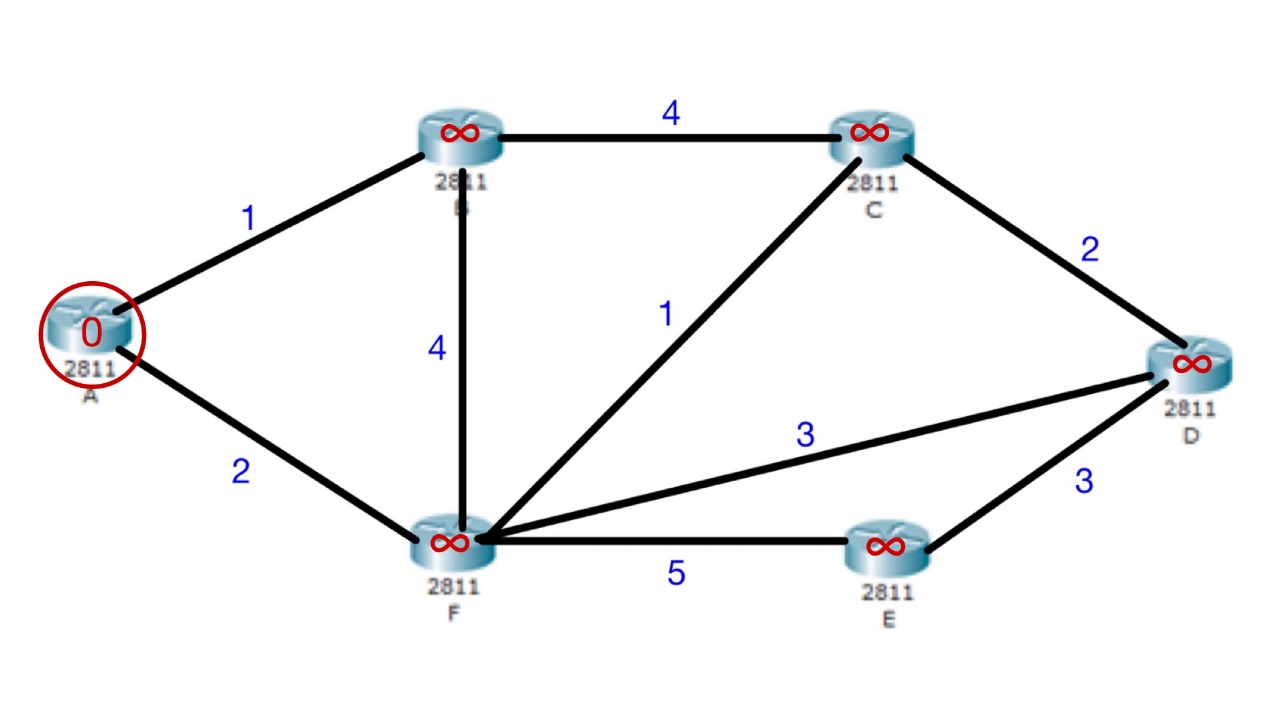


**问题分析与模型建立：**

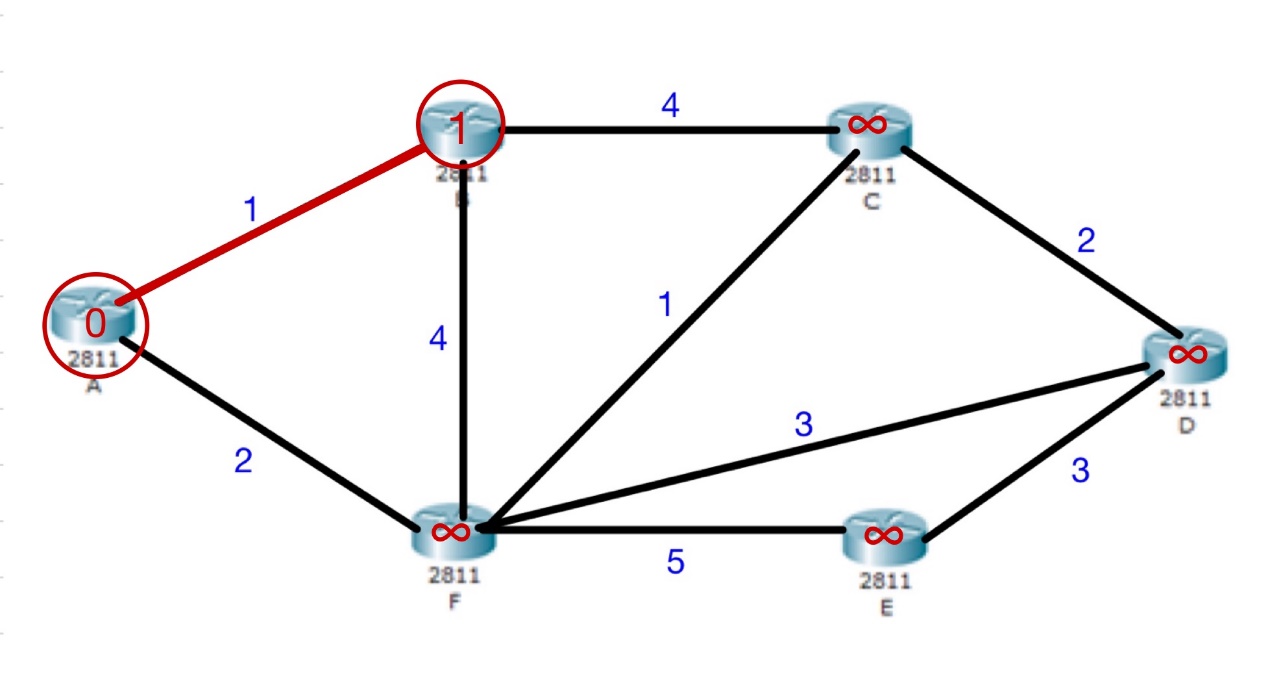
1. 根据图论的相关知识，对于给定的任意一点，存在着多条链路到底其他的各个点；我们在这里利用路由算法的分析寻找最短路径的线路计入指定点的一张表中，在路由器又称路由表。
2. 首先，先定义几个变量：
   1. 结点个数n；
   2. 二维矩阵M（nxn），距离矩阵，连通的结点间即为距离，不连通的结点间为正无穷，和自己的距离为0；
   3. 一维矩阵pb（1xn），若第i点已找到最短路径，则fp（i）=1，否则等于0，对于初始结点，fp=1；
   4. 距离矩阵d（1xn），若第i点已找到最短路径，则的d（i）=最短距离，否则为0，初始结点d=0；
   5. 上一结点矩阵path（1xn)，若第i点找到了最短路径，则path存放这一条最短路径的前一个结点，通过对每一点的回溯，可以找到最短路径。
3. 根据距离写出以下距离矩阵 ，这个矩阵一定是对称矩阵，因为A到B距离与B到A距离一定是一样的。



1. 我们拿路由器A举例分析：对于A自身，则pb(1)=1
2. 在图中，结点上，我们将已找到最短路径的点标为它的最短距离，（可以理解为A路由器已找到最短路径，距离为0），未找到的其余点表为正无穷（即表示不连通）。

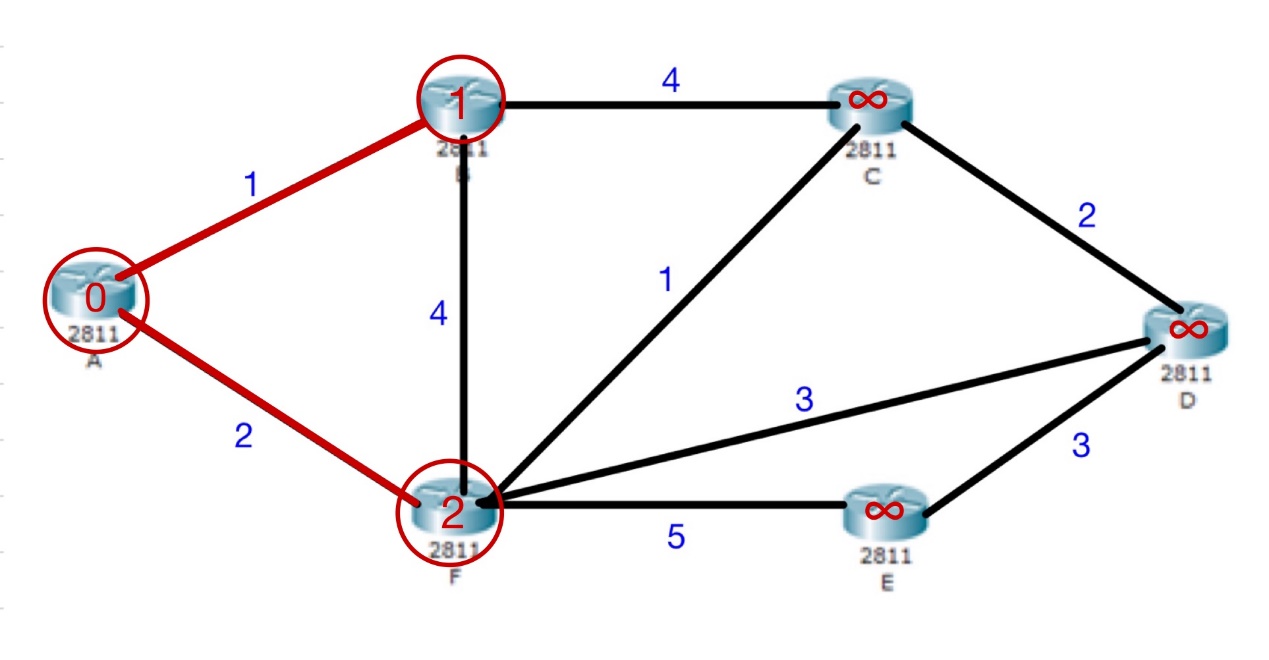


1. 在与A连通的点中，即在矩阵m的第1行，寻找最小值，最小值所在列即确定的最短路径的结点，如同B最短，pb（2）=1，d（2）=1，对于已找到最短路径的B上一结点为A，path（2）=1；



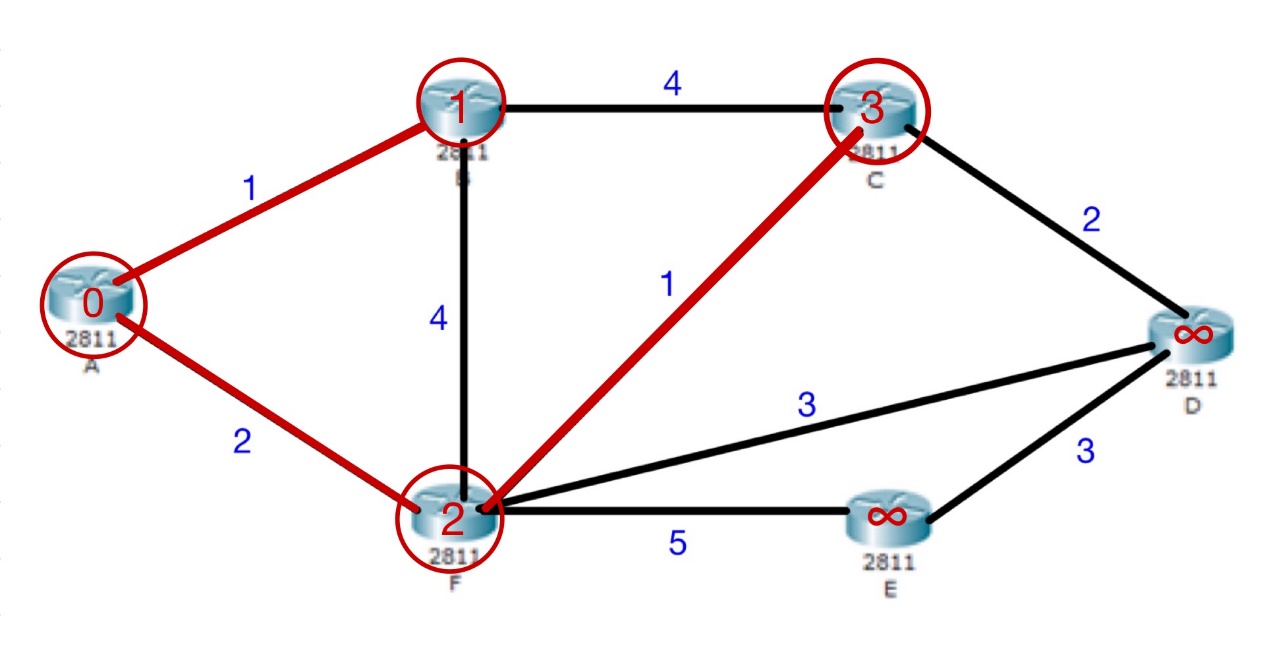
1. 接着，在
   1. 与A连通的，且未找到最短距离的节点的距离
   2. 与B连通的，且未找到最短距离节点的距离+B的最短距离

以上两种中寻找最短距离，最短为F，pb（6）=1；d（6）=2； path（6)=1;

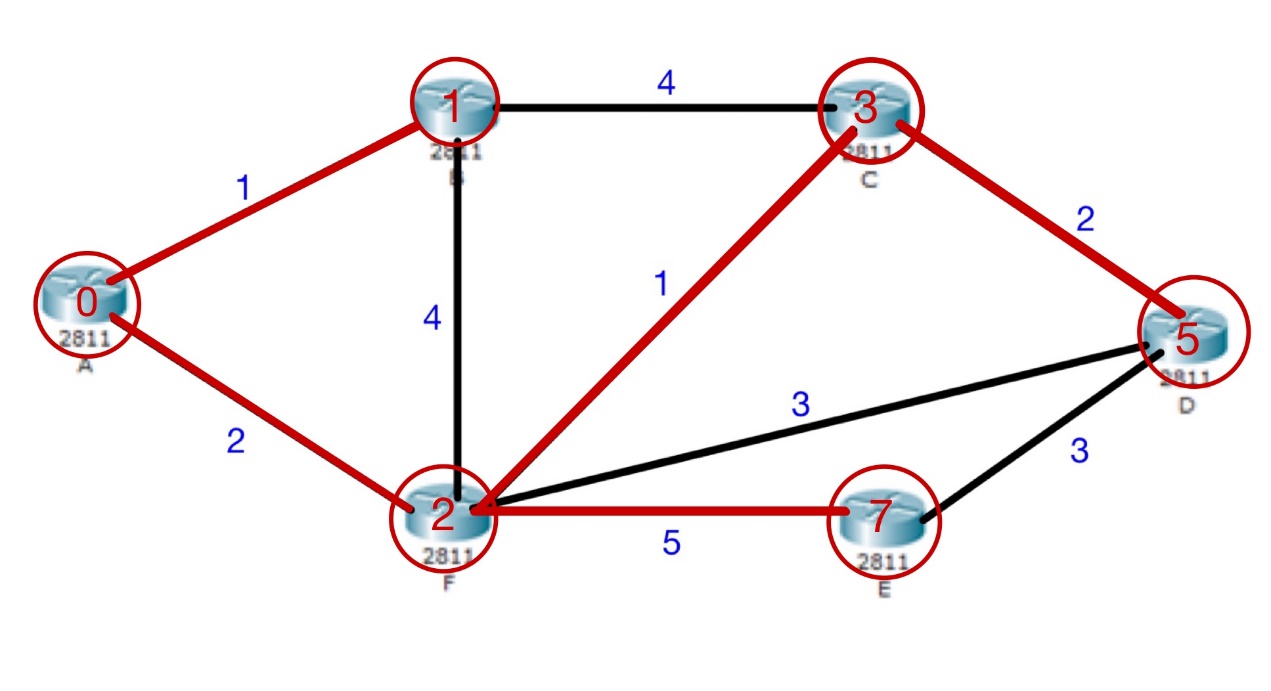


1. 重复以上步骤，在
   1. 与A连通的，且未找到最短距离的节点的距离
   2. 与B连通的，且未找到最短距离节点的距离+B的最短距离
   3. 与F连通的，且未找到最短距离节点的距离+F的最短距离

以上三种中寻找最短路径，最短为C,pb(3)=1;d(3)=3);path(3)=6;



1. 我们可以发现，所要寻找的最短路径即为
   1. 对于已找到最短路径的点（包括初始结点），在与其连通的，未找到最短路径的结点中，将之间距离与圆圈中的距离（即上一结点已找到的最短路径）相加，求得的最小值。
   2. 如果有多个相同的最短距离，任取其中一个。
   3. 最终最短路径即距离如下图:



**问题解决：**

1. 根据模型的基本成型后：我们开始利用MATLAB对路由表A的相关信息，即路由器A到达其他路由器的最短路径距离。
2. 通过过程分析代码解决主要包括三个部分：
   1. 距离矩阵的建立：

n=6; %设置距离矩阵的大小

temp=1; %设置起始点

m=zeros(6);%定义6阶零矩阵

m(1,2)=1;m(1,6)=2;%定义矩阵中相连存在实际距离的点

m(2,1)=1;m(2,3)=4;m(2,6)=4;

m(3,2)=4;m(3,4)=2;m(3,6)=1;

m(4,3)=2;m(4,5)=3;m(4,6)=3;

m(5,4)=3;m(5,6)=5;

m(6,1)=2;m(6,2)=4;m(6,3)=1;m(6,4)=3;m(6,5)=5;

for i=1:n %将不互通的点间距离置为无穷大

for j=1:n

if(m(i,j)==0)

m(i,j)=inf;

end

end

end

for i=1:n %将自己和自己相连的距离置为零

m(i,i)=0;

end

* 1. 最短路径算法的实现：

pb(1:length(m))=0;

pb(temp)=1;%求出最短路径的点为1，未求出的为0

d(1:length(m))=0;%存放各点的最短距离

path(1:length(m))=0;%存放各点最短路径的上一点标号

while sum(pb)<n %判断每一点是否都已找到最短路径

tb=find(pb==0);%找到还未找到最短路径的点

fb=find(pb);%找出已找到最短路径的点

min=inf;

for i=1:length(fb)

for j=1:length(tb)

plus=d(fb(i))+m(fb(i),tb(j)); %比较已确定的点与其相邻未确定点的距离

if((d(fb(i))+m(fb(i),tb(j)))<min)

min=d(fb(i))+m(fb(i),tb(j));

lastpoint=fb(i);

newpoint=tb(j);

end

end

end

d(newpoint)=min;

pb(newpoint)=1;

path(newpoint)=lastpoint; %最小值时的与之连接点

end

* 1. 展示运行结果：

d %展示A路由器到其他各个路由器

* 1. 全部代码：

n=6; %设置距离矩阵的大小

temp=1; %设置起始点

m=zeros(6);%定义6阶零矩阵

m(1,2)=1;m(1,6)=2;%定义矩阵中相连存在实际距离的点

m(2,1)=1;m(2,3)=4;m(2,6)=4;

m(3,2)=4;m(3,4)=2;m(3,6)=1;

m(4,3)=2;m(4,5)=3;m(4,6)=3;

m(5,4)=3;m(5,6)=5;

m(6,1)=2;m(6,2)=4;m(6,3)=1;m(6,4)=3;m(6,5)=5;

for i=1:n %将不互通的点间距离置为无穷大

for j=1:n

if(m(i,j)==0)

m(i,j)=inf;

end

end

end

for i=1:n %将自己和自己相连的距离置为零

m(i,i)=0;

end

pb(1:length(m))=0;

pb(temp)=1;%求出最短路径的点为1，未求出的为0

d(1:length(m))=0;%存放各点的最短距离

path(1:length(m))=0;%存放各点最短路径的上一点标号

while sum(pb)<n %判断每一点是否都已找到最短路径

tb=find(pb==0);%找到还未找到最短路径的点

fb=find(pb);%找出已找到最短路径的点

min=inf;

for i=1:length(fb)

for j=1:length(tb)

plus=d(fb(i))+m(fb(i),tb(j)); %比较已确定的点与其相邻未确定点的距离

if((d(fb(i))+m(fb(i),tb(j)))<min)

min=d(fb(i))+m(fb(i),tb(j));

lastpoint=fb(i);

newpoint=tb(j);

end

end

end

d(newpoint)=min;

pb(newpoint)=1;

path(newpoint)=lastpoint; %最小值时的与之连接点

end

d %展示A路由器到其他各个路由器

**结果分析和心得体会：**

1. **结果分析**

* 程序运行后结果：

d =

0 1 3 5 7 2

* 结果对应的表格转化：

|  |  |
| --- | --- |
| 与A路由器计算最短路径的路由器 | 最短距离 |
| A | 0 |
| B | 1 |
| C | 3 |
| D | 5 |
| E | 7 |
| F | 2 |

* 分析：我们拿了整个网络的一小部分为例，展示了路由器传送信息路径的基本选择方法。运行结果与问题分析结果一致，该算法基本符合了路由，将A路由器的路由表为例，其他路由器的路由表类似。

1. **心得体会**

通过结合计算机网络以及数学实验两门课程，我们实现了路由器的转发功能。在建模的过程中我们体会到了MATLAB在数据处理方面的优势：语法简单、丰富的库函数可供调用、调试过程简便等等。但是我们也发现， 算法时间复杂度未必最小。通过查阅资料我们发现还可以用MATLAB实现对路由器转发功能的核心思想——DIJKSTRA算法的改进。常见的方法是建立堆查找，可以更快的找到下一个要取的最小值节点(堆顶点),得到当前节点时候，扩展下一层节点时候，就修改移动堆中的值，这样每次只取堆的顶节点作为当前节点，不断的扩展下去，到取到目的节点就结束。通过这次实验我们能将理论与实际结合，在自己掌握的领域把MATLAB进行了实际应用，也是这八周课程一个最大的收获。

**参考文献：**

[1] 谢希仁.计算机网络[M].7版.北京:电子工业出版社,2017.1.

[2]冯晓东,梁红硕.路由算法的设计与优化[J].石家庄职业技术学院学报,2015,27(06):43-45.

[3]李萍.迪杰斯特拉算法的改进与实现[J].信息化建设,2016(02):345.