4.2

这题是一题典型的求最短路径问题，我们可以用到Dijkstra算法来求解

**代码如下：**

clc, clear;

L= {'A','B1',2;'A','B2',4 ;'B1','C1',3

'B1','C2',3;'B1','C3',1;'B2','C1',2

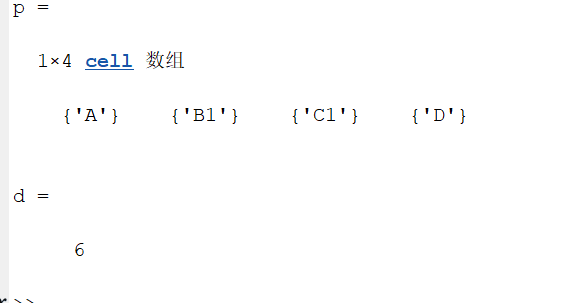
'B2','C2',3;'B2','C3',1;'C1','D',1

'C2','D',3 ;'C3','D',4};

G=digraph (L(:,1),L(:,2),cell2mat (L(:,3)));

plot (G), [p,d]=shortestpath(G,'A','D')

**结果如下：**



4.7

这题也是一题典型的求最短路径问题，同样的，我们可以用到Dijkstra算法来求解

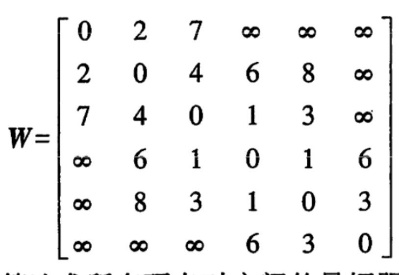
**符号说明**

：到j村所需要的距离

：j村有的小学生人数

**模型建立**

我们首先可以得到对应的邻接矩阵



然后我们可以调用Dijkstra算法来求出所有顶点对的距离，最后问题便转化为求



**代码如下：**

clc,clear,w=zeros(6);

w(1,[2,3])=[2,7];w(2,[3:5])=[4,6,8];

w(3,[4,5])=[1,3];w(4,[5,6])=[1,6];

w(5,6)=3;

G=graph(w,'Upper');

d=distances(G)

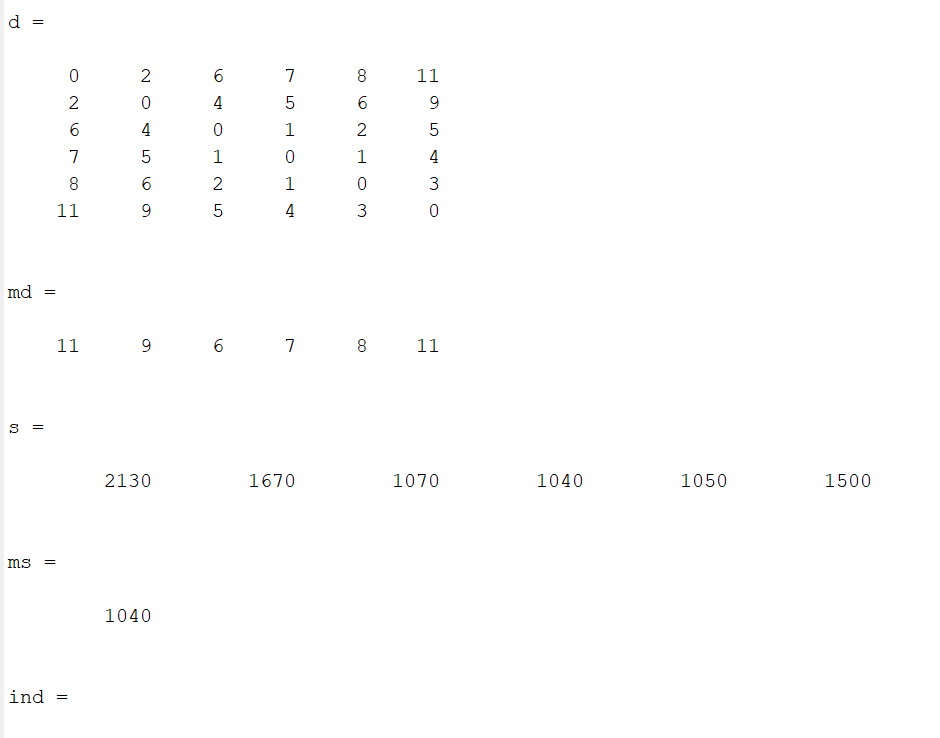
md=max(d)

c=[50 40 60 20 70 90];

s=c\*d

[ms,ind]=min(s)

**结果如下：**

结果

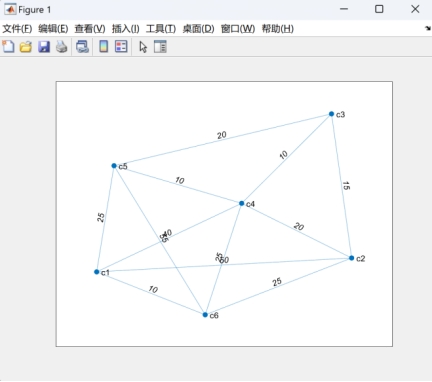
结果表明，选村庄四最好

补充题：

1. 问题分析

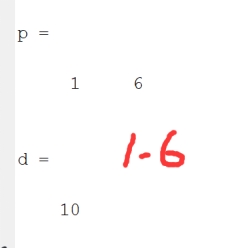
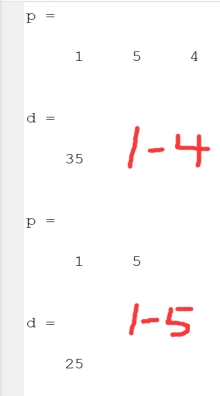
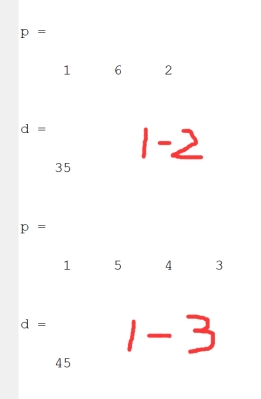
某公司在六个城市c1, c2, ... , c6中有分公司,从ci到cj,的直接航程票价记在下述矩阵的(i,j)位置上（∞表示无直接航路)。画出该矩阵对应的赋权图（顶点和边都要有标注），并帮助该公司设计一个简便的算法，能快速得到一张城市c1到其它城市间的票价最便宜的路线图。

1. 赋权图

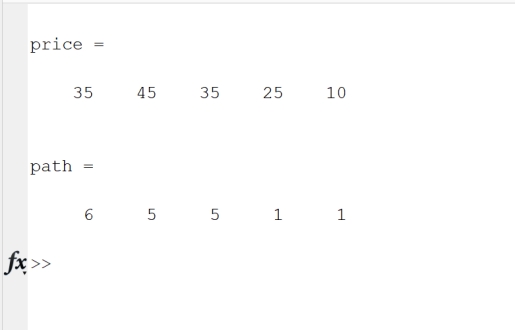


1. 从求解器确定答案：

从c1到其他城市的最短路径以及最短距离如下所示



4.从Floyd算法求解



有图可知:

c1 -c2:1-6-2 ,35

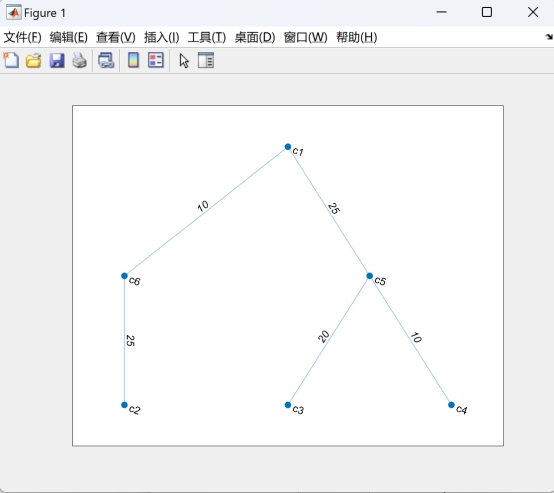
c1 -c3:1-5-3 ,45

c1 -c4:1-5-4 ,35

c1 -c5:1-5 ,25

c1 -c6:1-6 ,10

票价最便宜路线图



代码如下：

clc,clear,w = zeros(6);

w(1,[2,4,5,6]) = [50,40,25,10];w(2,[1,3,4,6]) = [50,15,20,25];

w(3,[2,4,5]) = [15,10,20];w(4,[1,2,3,5,6]) = [40,20,10,10,25];

w(5,[1,3,4,6]) = [25,20,10,55];w(6,[1,2,4,5]) = [10,25,25,55];

%构造完整的邻接矩阵

s = cellstr(strcat('c',int2str([1:6]')));%顶点字符串

G = graph(w,s,'upper');%利用邻接矩阵的上三角元素构造无向图

plot(G,'EdgeLabel',G.Edges.Weight);%画无向图

[price,path] = Floyd(w) %显示c1至其他地方票价最便宜价钱，路线矩阵

p = zeros(6);

p(1,5:6) = [25,10];p(5,3:4) = [20,10];p(6,2) = 25;

p = p+p';

G1 = graph(p,s,'upper');%利用邻接矩阵的上三角元素构造无向图

plot(G1,'EdgeLabel',G1.Edges.Weight);%画最短航线图

function [price,path] = Floyd(w)

%输出矩阵为两两顶点间最短距离矩阵，输入矩阵为待求邻接矩阵

n = length(w);

w(w == 0) = inf; %把零元素换成无穷大

w(1:n+1:end) = 0; %把对角线元素换成0

path = -ones(n);%初始化path矩阵

for i=1:n

for j=1:n

if i~=j && w(i,j)~=inf

path(i,j)=i;

end

end

end %完成path矩阵未更新的建立

for k = 1:n

for i = 1:n

for j = 1:n

if w(i,k)+w(k,j) < w(i,j) %Floyd算法核心，更新

w(i,j) = w(i,k)+w(k,j);

path(i,j) = k; %对w以及path矩阵更新

end

end

end

end

price = w(1,2:n);

path = path(1,2:n);%只考虑c1至其他城市，故只取部分