欢迎，规划类问题(运筹学)虽然结束了，但是呢，下面就进入了图与网络(图论)的内容了，欢迎进入新的篇章。
注意，本人学的书籍的版本有些古老,在最新版本的matlab中已经有很多图论工具箱中的函数不再适用。
请关注mathwork的官网以寻找最新版本适用的函数。

1 图与网络-简介

这里就可以引入很多东西了，就可以分享一下我以前看过的东西了；任何一种算法都会对应一种数据结构。例如二分查找对应的是顺序表（因为不可能在链表上执行二分查找）、递归对应的是树、最短路对应的是图。在动态规划那一节，你应该可以寻找到许多其他语言所做的题解；比如C和Java以及Python等。所以吧，计算机才是出路（bushi）。算法这东西是需要人去学习的；根据英雄哥的说法，算法出来的时候应该是思想，可以移植到其他系统或者语言的（扯远了）。

简单来说，图与网络就是就是一种思路，图论为任何一个包含了一种二元关系的离散系统提供了一个数学模型。

八股文：图与网络是运筹学（Operations Research）中的一个经典和重要的分支，所研究的问题涉及经济管理、工业工程、交通运输、计算机科学与信息技术、通讯与网络技术等诸多领域。下面将要讨论的最短路问题、最大流问题、最小费用流问题和匹配问题等都是图与网络的基本问题。

1.1 简单的图示，网络流规划

例题：

某公司在六个城市 $c_1, c_2, c_3, c_4, c_5, c_6$ 中有分公司，从 c_i 到 c_j 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），请帮助该公司设计一张城市 c_1 到其它城市间的票价最便宜的路线图。

$$\begin{bmatrix} 0 & 50 & \infty & 40 & 25 & 10 \\ 50 & 0 & 15 & 20 & \infty & 25 \\ \infty & 15 & 0 & 10 & 20 & \infty \\ 40 & 20 & 10 & 0 & 10 & 20 \\ 25 & \infty & 20 & 10 & 0 & 55 \\ 10 & 25 & \infty & 25 & 55 & 0 \end{bmatrix}$$

2 图与网络-适用题目

2.1 图与网络适用的赛题：

分很多类，这里简单讲述一下:大概就是SSP,TSP一类问题，这个时候穷举法能找到最优解，但是耗时太长，可以使用图与网络知识快速求出局部最优解。

·指派问题 一家公司经理准备安排 N 名员工去完成 N 项任务，每人一项。由于各员工的特点不同，不同的员工去完成同一项任务时所获得的回报是不同的。如何分配工作方案可以使总回报最大?

·运输问题 某种原材料有 M 个产地，现在需要将原材料从产地运往 N 个使用这些原材料的工厂。假定 M 个产地的产量和 N 家工厂的需要量已知，单位产品从任一产地到任一工厂的运费已知，那么如何安排运输方案可以使总运输成本最低?

·最短路径 一名货柜车司机奉命在最短的时间内将一车货物从甲地运往乙地。从甲地到乙地的公路网纵横交错，因此有多种行车路线，这名司机应选择哪条线路呢？假设货柜车的运行速度是恒定的，那么这一问题相当于需要找到一条从甲地到乙地的最短路。

上述问题有两个共同的特点：一是它们的目的都是从若干可能的安排或方案中寻求某种意义下的最优安排或方案，数学上把这种问题称为最优化或优化（optimization）问题；二是它们都易于用图形的形式直观地描述和表达，数学上把这种与图相关的结构称为网络（network）。与图和网络相关的最优化问题就是网络最优化或称网络优化（netwok optimization）问题。

2.2 模型讲解

2.2.1 图的基本概念

1. 如下，一个图主要由节点v_i，边(弧)e_j等组成；也就是说，从定义来讲，一个无向(有向)图(undirected graph)G 是由一个非空有限集合$V(G)$ 和$V(G)$ 中某些元素的无序(有序)对集合 $E(G)$ 构成的二元组，记为$G = (V(G), E(G))$ 。其中 $V(G) = \{v_1, v_2, \cdots, v_i\}$ 称为图G 的顶点集（vertex set）或节点集（node set），V(G) 中的每一个元素 $v_i(i = 1, 2, , n)$ 称为该图的一个顶点（vertex）或节点（node）； $E(G) = \{e_1, e_2, \cdots, e_j\}$ 称为图G 的边(弧)集（edge set），E(G) 中的每一个元素 $e_j(j = 1, 2, \cdots, m)$ (即V(G) 中某两个元素 v_i, v_j 的无序(有序)对) 记为 $e_k(a_k) = (v_i, v_j)$ 或 $e_k(a_k) = v_iv_j = v_jv_i(k = 1, 2, \cdots, m)$ ，被称为该图的一条从 v_i 到 v_j 的边(弧)（edge）。

2. 当边 $e_k = v_iv_j$ 时，称 v_i, v_j 为边 e_k 的端点，并称 v_i 与 v_j 相邻（adjacent）；边 e_k 称为与顶点 v_i, v_j 关联（incident）。如果某两条边至少有一个公共端点，则称这两条边在图G 中相邻。一个图称为有限图，如果它的顶点集和边集都有限。图G的顶点数用符号 $|V|$或$v(G)$表示，边数用$|E|$或$\varepsilon(G)$表示。

当弧$a_k = v_iv_j$时,称v_i是弧a_k的尾,v_i为弧的头，并称弧a_k为v_i的出弧,v_j的入弧。

对于有向图D，可以在相同顶点集上作一个图G，使得对于 D 的每条弧，G有一条有相同端点的边与之相对应。这个图称为 D 的基础图。

给定任意图G，对于它的每个边，给其端点指定一个顺序，从而确定一条弧，由此得到一个有向图，这样的有向图称为G的一个定向图。

简单记录边与弧的差别：有向边也称弧,边的始点称为弧尾,终点称为弧头。

只有一个顶点的图称为平凡图，其他的所有图都称为非平凡图。

端点重合为一点的边称为环(loop)。

重边与自环请见下图。

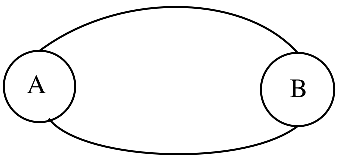


图1.重边

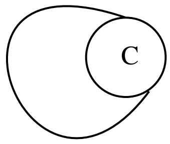


图2.自环

3. 简单图和非简单图示例：

简单图就是没有重边或者自环的图形。

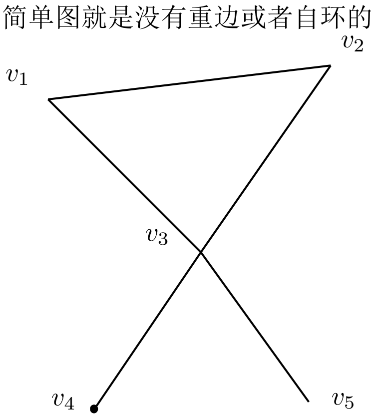


图1.简单图

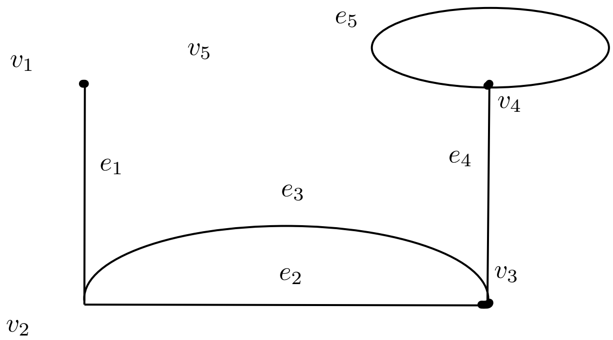


图2.非简单图

4. 有向图与无向图的示例：

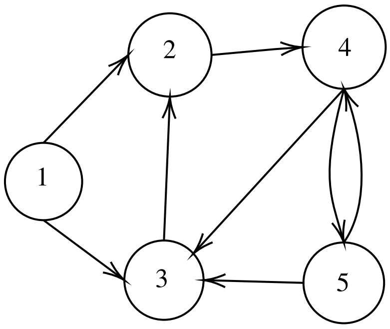


图1.有向图

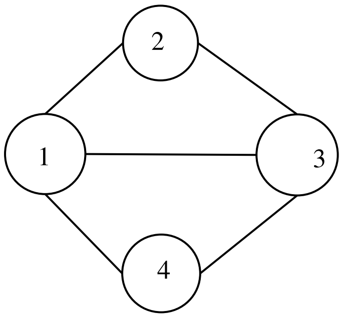


图2.无向图

5. 图的矩阵表示:

1).邻接矩阵表示法:对于两个节点之间，如果有一条弧相连就记为1。对于邻接矩阵$C=\{c_{ij}\}_{n\times n}=\{0,1\}^{n\times n}$

也就是说 $c_{ij}=\begin{cases}1 & (i,j)\in A\\0 & (i,j)\notin A\end{cases}$

那么我们可以简单的写出4.图1.有向图里面的邻接矩阵:

$$C=\begin{bmatrix}0&1&1&0&0\\0&0&0&1&0\\0&1&0&0&0\\0&0&1&0&1\\0&0&1&1&0\end{bmatrix}$$

然后再写出4.图2无向图里面的邻接矩阵:

$$C=\begin{bmatrix}0&1&1&1\\1&0&1&0\\1&1&0&1\\1&0&1&0\end{bmatrix}$$

这里有个需要注意的地方：无向图的邻接矩阵一定是对称的，所以后面Matlab画图的时候需要选择下三角或上三角矩阵。

2).关联矩阵表示法：对于两个节点之间如果这个点是弧的起点，则这个点记为1；如果这个点是弧的终点，则这个点记为-1；如果一个弧和一个节点不关联，则这个点记为0。对于关联矩阵$B=\{b_{ij}\}_{n\times m}=\{-1,0,1\}^{n\times m}$

注意：关联矩阵的每一行表示节点，每一列表示弧。

那么我们可以简单的写出4.图1.有向图里面的关联矩阵，首先假设对应的弧分别是(1,2)，(1,3)，(2,4)， (3,2)，(4,3)，(4,5)，(5,3)和(5,4)，则这个有向图的关

联矩阵：

$$B=\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 \end{bmatrix}$$

通过观察，我们可以给出一个有向图关联矩阵的小结论：每一列只有两个非零元(-1,1)，同时可以有任意n-1列推出剩下的第n列。

同理，我们可以简单写出4.图2无向图里面的关联矩阵,首先可以看出这个时候每个端点跟边都是关联的，假设按照(1,2),(2,3),(3,4),(4,1),(3,1)顺序表示边，则这个无向图的关联矩阵：

$$B=\begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

与有向图的结论相似，每一列都会有两个非零元(1,1)；重边对应的列元素完全一样。

此外，还有弧表表示法，邻接表表示法，星形表示法等不赘述。

还可以补充一下轨，迹和连通的概念。$W=v_0e_1v_1e_2\cdots e_kv_k$,其中$e_i\in E(G),1\leq i\leq k,v_j\in V(G),0\leq j\leq k,e_i$与$v_{i-1},v_i$关联，称$W$是图$G$的一条道路，k为路长，顶点$v_0$与$v_k$分别称为$W$的起点和终点，而$v_1,v_2,\cdots,v_{k-1}$称为它的内部顶点。如果道路$W$的边互不相同，则称$W$为迹；如果道路$W$的顶点互不相同，则称$W$为轨。如果图$G$的两个顶点$u,v$之间存在道路，则称$u$和$v$连通。$u,v$之间最短的轨迹长度叫做$u,v$之间的距离。记作$d(u,v)$。如果图$G$的任两个顶点都是连通的，则称$G$为连通图。

2.2.2 Matlab实现-matlab图论工具箱

对于绘制无向图，Matlab提供了以下较为常用命令：
创建空的无向图对象
G = graph;
使用邻接矩阵A创建赋权无向图
G = graph(A);
使用邻接矩阵A和顶点nodes创建赋权无向图，其中nodes是表示顶点的字符串
G = graph(A, nodes);
使用顶点对创建无向图
G = graph(s, t);
使用顶点对s.t和权重向量创建赋权无向图
G = graph(s, t, weight);
使用顶点对s.t和权重向量创建赋权无向图，并使用字符向量元胞数组nodes指定顶点名称
G = graph(s, t, weight, nodes);
使用数值标量num指定图中的节点数
G = graph(s, t, weights, num);

对于绘制有向图，Matlab提供了以下较为常用命令：
创建空的有向图对象
G = digraph;
使用邻接矩阵A创建赋权有向图
G = digraph(A);
使用邻接矩阵A和节点名称nodenames创建有向图
G = digraph(A, nodenames);
使用顶点对s.t创建有向图
G = digraph(s, t);
使用顶点对s.t和权重向量创建赋权有向图
G = digraph(s, t, weights);
使用顶点对s.t和权重向量创建赋权有向图，并使用字符向量元胞数组nodes指定顶点名称
G = digraph(s, t, weights, nodenames);
使用顶点对s.t和权重向量创建赋权有向图，并使用节点表NodeTable
G = digraph(s, t, weights, NodeTable);
使用数值标量num指定图中的节点数
G = digraph(s, t, weights, num);

强烈推荐去Mathwork官网看这个工具箱，这里给出一些简略的讲述和例子。

绘制如下无向图,分别通过邻接矩阵，节点向量绘制无向图，随后再给出每条边的权重,并给出指定节点名称的代码

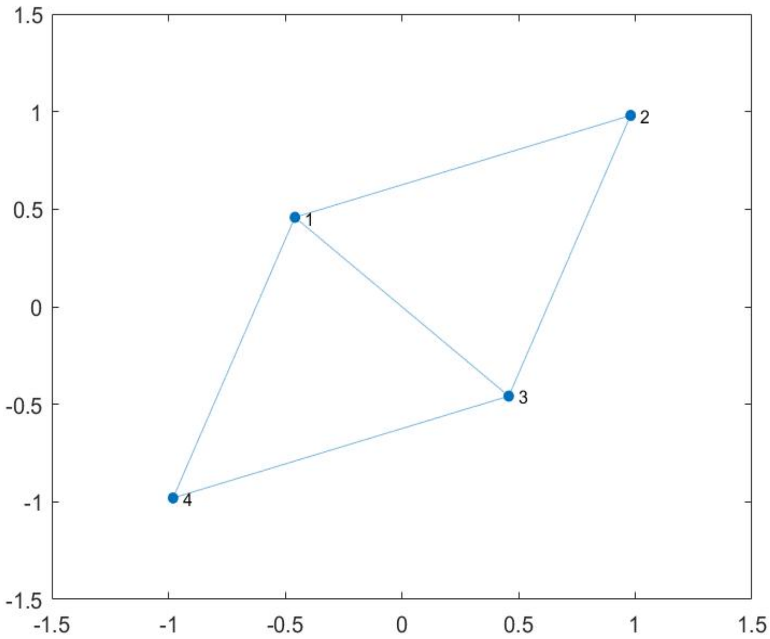


图 1. 示例无向图

代码 1. FirstExample.m

```
        %This is  example1.m
A = [0, 1, 1, 1; ...
     1, 0, 1, 0; ...
     1, 1, 0, 1; ...
     1, 0, 1, 0;];
G = graph(A);
plot(G);
        %This is  example2.m
s = [1, 2, 3, 3, 4];
t = [2, 3, 4, 1, 1];
G = graph(s, t);
plot(G);
        %This is  example3.m
s = [1, 2, 3, 3, 4];
t = [2, 3, 4, 1, 1];
w = [9, 8, 7, 5, 2];
G = graph(s, t, w);
```

```
plot(G, 'EdgeLabel', G.Edges.Weight, 'linewidth', 2);
%Please read Mathwork graph-plot for more information.
        %This is example4.m
s = [1, 2, 3, 3, 4];
t = [2, 3, 4, 1, 1];
w = [9, 8, 7, 5, 2];
nodesname = {'Alpha', 'Beta', 'Gamma', 'Delta'};
G = graph(s, t, w, nodesname);
plot(G, 'EdgeLabel', G.Edges.Weight, 'linewidth', 2);
set(gca, 'XTick', [], 'YTick', []);
```

2.2.3 树的基本概念

6.树,一个树类似于一个连通的无向图。

当且仅当一幅含有V个结点的图G满足下列五个条件之一时，它就是一棵树：

G有V-1条边且不含有环

G有V-1条边且是连通的

G是连通的，但删除任意一条边都会使它不再连通

G是无环图，但添加任意一条边都会产生一条环

G中的任意一对顶点之间仅存在一条简单路径。

树的定义有些许抽象，我们来点实际例子。

连线问题的数学模型是在连通赋权图上求权最小的生成树。赋权图的具最小权的生成树叫做最小生成树。主要有prim 算法构造最小生成树与 Kruskal 算法构造最小生成树。同时还可以用这两个算法求解。

2.2.4 Matlab实现-matlab生成树

matlab生成树提供的函数有：

返回图G生成的最小生成树，(Name与Value用来枝定生成算法)

T = minspantree(G,Name,Value);

从一个源节点到另一个节点的最短路径树

TR = shortestpathtree(G,s,t,Name,Value);

代码 2. SecondExample.m

```
        %This is example1.m
A = [0, 1, 1, 0, 0, 0, 0; ...
     1, 0, 0, 1, 1, 0, 0; ...
     1, 0, 0, 1, 0, 0, 1; ...
```

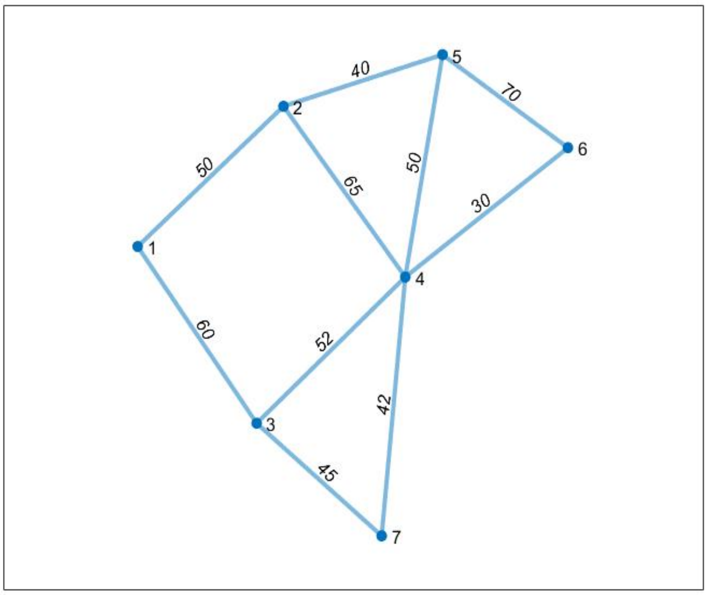


图 **2.** 示例生成树

```
      0, 1, 1, 0, 1, 1, 1; ...
      0, 1, 0, 1, 0, 1, 0; ...
      0, 0, 0, 1, 1, 0, 0; ...
      0, 0, 1, 1, 0, 0, 0];
G = graph(A);
G.Edges.Weight(1) = 50;
G.Edges.Weight(2) = 60;
G.Edges.Weight(3) = 65;
G.Edges.Weight(4) = 40;
G.Edges.Weight(5) = 52;
G.Edges.Weight(6) = 45;
G.Edges.Weight(7) = 50;
G.Edges.Weight(8) = 42;
G.Edges.Weight(8) = 30;
G.Edges.Weight(9) = 42;
G.Edges.Weight(10) = 70;
plot(G, 'EdgeLabel', G.Edges.Weight, 'linewidth', 2)
            %This is  example2.m
```

```
clc;
clear;
a = zeros(7);
a(1, 2) = 50;a(1, 3) = 60;
a(2, 4) = 65;a(2, 5) = 40;
a(3, 4) = 52;a(3, 7) = 45;
a(4, 5) = 50;a(4, 6) = 30;a(4, 7) = 42;
a(5, 6) = 70;
a = a + a';
a(find(a == 0)) = inf;
result = [];
p = 1;
tb = 2:length(a);
while length(result) ~= length(a) - 1
    temp = a(p, tb);
    temp = temp(:);
    d = min(temp);
    [jb, kb] = find(a(p, tb) == d);
    j = p(jb(1));
    k = tb(kb(1));
    result = [result, [j; k; d]];
    p = [p, k];
    tb(find(tb == k)) = [];
end
result
            %This is example3.m
clc;
clear;
a(1, 2) = 50;a(1, 3) = 60;
a(2, 4) = 65;a(2, 5) = 40;
a(3, 4) = 52;a(3, 7) = 45;
a(4, 5) = 50;a(4, 6) = 30;a(4, 7) = 42;
a(5, 6) = 70;
[i, j, b] = find(a);
data = [i'; j'; b'];
index = data(1:2, :);
loop = max(size(a)) - 1;
result = [];
```

```
while length(result) < loop
    temp = min(data(3, :));
    flag = find(data(3, :) == temp);
    flag = flag(1);
    v1 = data(1, flag);
    v2 = data(2, flag);
    if index(1, flag) ~= index(2, flag)
        result = [result, data(:, flag)];
    end
    index(find(index == v2)) = v1;
    data(:, flag) = [];
    index(:, flag) = [];
end
s = result(1, :);
t = result(2, :);
w = result(3, :);
G = graph(s, t, w);
plot(G, 'EdgeLabel', G.Edges.Weight, 'linewidth', 2);
result
```

2.3 图与网络示例

例子1.求图3.中v_1到v_{11}的最短路径。可以使用graphshortestpath函数来寻找最短的距离，并且给出沿着这个最短路径的各个节点。

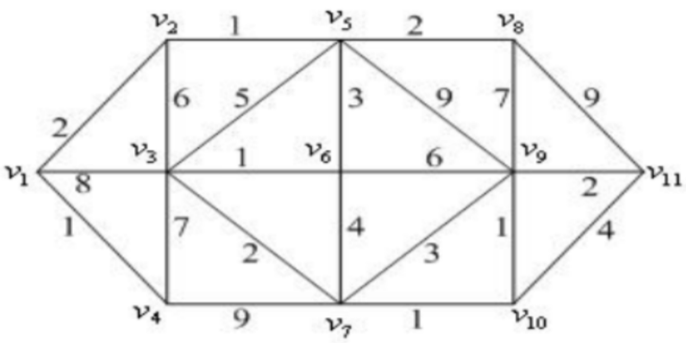


图 **3.** 无向图的最短路径

代码 **3.** Example1.m

```
%This is example1.m
clc,clear;
a(1,2)=2;a(1,3)=8;a(1,4)=1;
a(2,3)=6;a(2,5)=1;
a(3,4)=7;a(3,5)=5;a(3,6)=1;a(3,7)=2;
a(4,7)=9;
a(5,6)=3;a(5,8)=2;a(5,9)=9;
a(6,7)=4;a(6,9)=6;
a(7,9)=3;a(7,10)=1;
a(8,9)=7;a(8,11)=9;
a(9,10)=1;
a(9,11)=2;
a(10,11)=4;
a=a';
[i,j,v]=find(a);
b = sparse(i,j,v,11,11);
[x,y,z]=graphshortestpath(b,1,11,'Directed',false);
disp('The shortest distance = ');
disp(x);% 13
disp('The path =');
disp(y);
% 1 2 5 6 3 7 10 9 11
```

例子2.求图4.有向图v_s到v_t的最短路径及长度。

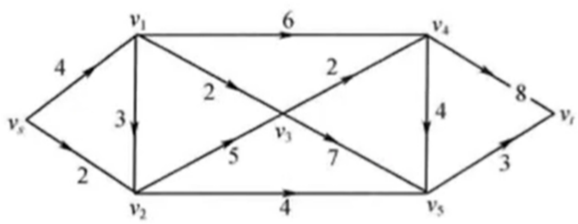


图 **4.** 有向图的最短路径

代码 **4.** Example2.m

```
%This is example2.m
```

```
clc,clear;
a=zeros(7);
a(1,2)=4;a(1,3)=2;
a(2,3)=3;a(2,4)=2;a(2,5)=6;
a(3,4)=5;a(3,6)=4;
a(4,5)=2;a(4,6)=7;
a(5,6)=4;a(5,7)=8;
a(6,7)=3;
b=sparse(a);
[x,y,z]=graphshortestpath(b,1,7,'Directed',1,'Method','Bellman-Ford');
view(biograph(b ,[]));
```

例子3.如下表，设有九个节点$v_i(i = 1, 2, \cdots, 9)$,坐标分别为(x_i, y_i),具体数据见下表。任意两个节点之间的距离为

$$d_{ij} = |x_i - x_j| + |y_i - y_j|$$

。问怎样连接电缆，使每个节点都连通，且所用的总电缆长度为最短?

表 1. 点的数据坐标

i	1	2	3	4	5	6	7	8	9
x_i	0	5	16	20	33	23	35	25	10
y_i	15	20	24	20	25	11	7	0	3

代码 5. Example3.m

```
        %This is  example3.m
clc,clear;
x=[0,5,16,20,33,23,35,25,10];
y=[15,20,24,20,25,11,7,0,3];
xy=[x;y];
d=mandist(xy);
d=tril(d);
b=sparse(d)
[ST,pred]=graphminspantree(b,'Method','Kruskal');
st=full(ST);
TreeLength=sum(sum(st))
view(biograph(ST,[],'ShowArrows','off'));
```

例子4.求下图中从①到⑧的最大流。

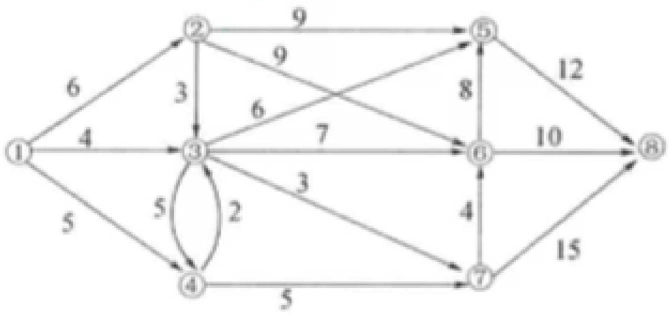


图 5. 最大流问题的网络图

代码 6. Example4.m

```
%This is example4.m
clc,clear;
a=zeros(9);
a(1,2)=6;a(1,3)=4;a(1,4)=5;
a(2,3)=3;a(2,5)=9;a(2,6)=9;
a(3,4)=4;a(3,5)=6;a(3,6)=7;a(3,7)=3;
a(4,7)=5;a(4,9)=2;
a(5,8)=12;
a(6,5)=8;a(6,8)=10;
a(7,6)=4;a(7,8)=15;
a(9,3)=2;
b=sparse(a);
[x,y,z]=graphmaxflow(b,1,8)
```

3 图与网络-代码实现

贴出matlab代码求解,请结合前面整理后的式子来分别对应变量

代码 7. Picture and Network

```
clc,clear
a=zeros(6);
a(1,2)=50;a(1,4)=40;a(1,5)=25;a(1,6)=10;
```

```
a(2,3)=15;a(2,4)=20;a(2,6)=25;
a(3,4)=10;a(3,5)=20;
a(4,5)=10;a(4,6)=25;
a(5,6)=55;
a=a+a';
a(find(a==0))=inf;
pb(1:length(a))=0;pb(1)=1;index1=1;index2=ones(1,length(a));
d(1:length(a))=inf;d(1)=0;temp=1;
while sum(pb)<length(a)
 tb=find(pb==0);
 d(tb)=min(d(tb),d(temp)+a(temp,tb));
 tmpb=find(d(tb)==min(d(tb)));
 temp=tb(tmpb(1));
 pb(temp)=1;
 index1=[index1,temp];
 temp2=find(d(index1)==d(temp)-a(temp,index1));
 index2(temp)=index1(temp2(1));
end
d, index1, index2
```

4 图与网络-实战演练

国赛很需要图与网络去做一些特殊解法，同时图与网络也是其他算法所必须要学习的东西，推荐进行学习。做题的话可以看

References

[1] 谢中华. MATLAB与数学建模[A].北京航空航天大学出版社[M]:科学技术协会,2021-02-14.

[2] 数学建模老师. 第9讲 图与网络模型及方法（一）[M]:https://www.bilibili.com/video/BV1na411Y7kP/?spm_id_from=333.337.search-card.all.click&vd_source=1db675367482476693cd8659d026e5b7,2022-04-25.

[3] 英雄哪里出来. 算法学习路线[A].https://ufbva3m5zn.feishu.cn/docs/doccnIQ95UPr5yabjQg2bFHl79f,2023-7-23.

[4] letterMe.第二章–网络与图(复杂网络学习笔记)[A].https://www.cnblogs.com/GGTomato/p/12654337.html, 2020-04-07 .

[5] Miao_Guo.图的一些基本知识：图，邻居，度矩阵，邻接矩阵[B].https://blog.csdn.net/luzaijiaoxia0618/article/details/104718146,2020-03-07.

[6] Miao_Guo.图的一些基本知识：关联矩阵、拉普拉斯矩阵[B].https://blog.csdn.net/luzaijiaoxia0618/article/details/104720948,2020-03-08.

[7] HachiLin.图论基础知识(二) —— 路与连通[B]https://blog.csdn.net/Hachi_Lin/article/details/88047555, 2019-02-28.

[8] Mathwork.(graph)具有无向边的图[M].https://ww2.mathworks.cn/help/matlab/ref/graph.html,

[9] Mathwork.(digraph)具备有向边的图[M].https://ww2.mathworks.cn/help/matlab/ref/digraph.html,

[10] Better-ing.matlab 绘制有向图、无向图、有权有向图、有权无向图以及查找最短路径[B].https://blog.csdn.net/weixin_43404836/article/details/114252343,2021-03-01.

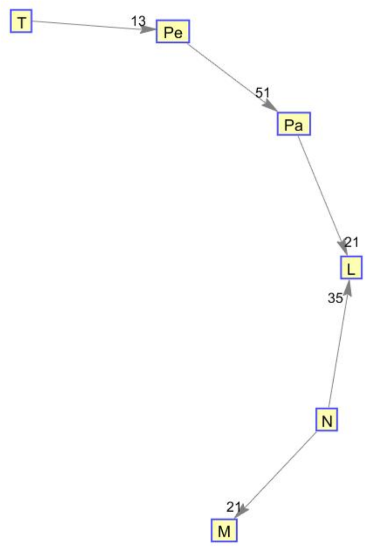
[11] Mathwork.GraphPlot 属性-图论图的外观和行为[M].https://ww2.mathworks.cn/help/matlab/ref/matlab.graphics.chart.primitive.graphplot-properties.html,

[12] MARK_REAPER.Matlab图论工具箱的利用[B].https://www.cnblogs.com/markReaper/p/8454817.html,2018-02-20 .

5 数学建模算法大全第五章习题答案

1. 简单思考可知，先运羊，接着随意带狼(菜)到对岸，再把羊带回去，再把菜(狼)带对岸，最后带羊回去即可。

2. 简单可见图如下：



3. 第二年处和第三年初都换一台新机器，总费用4万元。

4. 从仓库运往市场的最大流量为110单位，其中市场3只能满足50单位，差10单位。

5. 甲未得到聘用，乙-俄，丙-日，丁-英，戊-德。

6. 最小费用为240。