图论中最短路算法与程序实现

图论中的最短路问题(包括无向图和有向图)是一个基本且常见的问题。主要的算法有Dijkstra算法和Floyd算法。

Dijkstra算法是求出指定两点之间的最短路，算法复杂度为 $O(n^2)$

Floyd算法是求出任意两点之间的最短路，算法复杂度为 $O(n^3)$

1.Dijkstra算法

假定给出一个网络 $N=(V,E,W)$，现在要求出任意点 i 到任意点 j 之间的最短路，

算法描述如下：

(1) 给出初始点集合 $P=\{i\}$，剩余点集合 $Q=\{1,2,...,n\}-P$。初始 i 点到各点的直接距离 $U_r=w_{ir}(r=1,2,..,n)$。

(2) 在 Q 中寻找到 i 点距离最小的点 k，使得 $U_k=\min\limits_{r\in Q}\{U_r\}$，并置

$$P\bigcup\{k\}\to P,\quad Q-\{k\}\to Q$$

(3) 对 Q 中每个 r，如果 $U_k+w_{kr}<U_r$，则 $U_k+w_{kr}\to U_r$

然后返回(2)直到找到 j 点为止。

该算法经过 $n-1$ 次循环结束。在整个算法过程中，步骤(2)最多作 $\frac{1}{2}(n-1)(n-2)$ 次比较，步骤(3)最多作 $\frac{1}{2}(n-1)(n-2)$ 次加法和比较，因此总的计算量是 $O(n^2)$ 阶。

2. Floyd算法

1) 根据已知的部分节点之间的连接信息，建立初始距离矩阵 $B(i,j)$，其中没有给出距离的赋予一个充分大数值，以便于更新。$(i,j=1,2,\cdots.n)$

2)进行迭代计算。对任意两点 (i,j)，若存在 k，使 $B(i,k)+B(k,j)<B(i,j)$，则更新 $B(i,j)=B(i,k)+B(k,j)$

3)直到所有点距离不再更新停止计算。得到最短路距离矩阵 $B(i,j)$

算法程序(Matlab)为：

```
for k=1:n
for i=1 :n
   for j=1:n
      t=B(i,k)+B(k,j);
      if t<B(i,j)  B(i,j)=t; end
      end
   end
end
```

实例：

已知50个点之间相互连接信息见表1及续表。求最短距离矩阵

表1 各点距离(m)

起点	终点	距离	起点	终点	距离	起点	终点	距离
1	2	400	7	18	160	15	17	250
1	3	450	8	9	200	16	17	140
2	4	300	8	15	285	16	18	130
2	21	230	9	10	180	17	27	240
2	47	140	10	11	150	18	19	204
3	4	600	10	15	160	18	25	180
4	5	210	11	12	140	19	20	140
4	19	310	11	14	130	19	24	175
5	6	230	12	13	200	20	21	180
5	7	200	13	34	400	20	24	190
6	7	320	14	15	190	21	22	300
6	8	340	14	26	190	21	23	270
7	8	170	15	16	170	21	47	350

续表1 各点距离(m)

起点	终点	距离	起点	终点	距离	起点	终点	距离
22	44	160	22	29	31	36	40	190
22	45	270	22	30	31	37	38	135
22	48	180	22	30	42	38	39	130
23	24	240	23	30	43	39	41	310
23	29	210	23	31	32	40	41	140
23	30	290	23	31	36	40	50	190
23	44	150	23	31	50	42	50	200
24	25	170	24	32	33	43	44	260
24	28	130	24	32	35	43	45	210
26	27	140	26	32	36	45	46	240
26	34	320	26	33	34	46	48	280
27	28	190	27	35	37	48	49	200
28	29	260	28	36	39			

```
n=50;  %Matlab实现的Floyd算法
A=zeros(n,n);
for i=1:n
   for j=1:n
     if(i==j) A(i,j)=0;
     else A(i,j)=100000;
     end
   end
end  %赋直接距离信息
A(1,2)=400;A(1,3)=450; A(2,4)=300;A(2,21)=230; A(2,47)=140;A(3,4)=600;
A(4,5)=210;A(4,19)=310;A(5,6)=230;A(5,7)=200;  A(6,7)=320; A(6,8)=340;
A(7,8)=170;A(7,18)=160;A(8,9)=200;A(8,15)=285; A(9,10)=180; A(10,11)=150;
A(10,15)=160; A(11,12)=140; A(11,14)=130; A(12,13)=200; A(13,34)=400;
A(14,15)=190;A(14,26)=190; A(15,16)=170;  A(15,17)=250; A(16,17)=140;
```

```
A(16,18)=130; A(17,27)=240; A(18,19)=204; A(18,25)=180; A(19,20)=140;
A(19,24)=175; A(20,21)=180; A(20,24)=190; A(21,22)=300; A(21,23)=270;
A(21,47)=350;A(22,44)=160;A(22,45)=270;A(22,48)=180;A(23,24)=240;
A(23,29)=210;A(23,30)=290;A(23,44)=150;A(24,25)=170;A(24,28)=130;
A(26,27)=140;A(26,34)=320;A(27,28)=190;A(28,29)=260;A(29,31)=190;
A(30,31)=240;A(30,42)=130;A(30,43)=210;A(31,32)=230;A(31,36)=260;
A(31,50)=210;A(32,33)=190;A(32,35)=140;A(32,36)=240;A(33,34)=210;
A(35,37)=160;A(36,39)=180;A(36,40)=190;A(37,38)=135;A(38,39)=130;
A(39,41)=310;A(40,41)=140;A(40,50)=190;A(42,50)=200;A(43,44)=260;
A(43,45)=210;A(45,46)=240;A(46,48)=280;A(48,49)=200;
for j=1:n
   for i=1:j-1
      A(j,i)=A(i,j);  %使矩阵对称
   end
end
```

```
B=A;
%利用Floyd算法计算最短距离矩阵
for k=1:n
  for i=1 :n
     for j=1:n
          t=B(i,k)+B(k,j);
        if t<B(i,j)  B(i,j)=t; end
     end
  end
end
```

```
%输出距离矩阵到文件
fid=fopen('distance.txt','w');
  for i=1:n
     for j=1:n
          fprintf(fid,'%4d ',B(i,j));
     end
     fprintf(fid,'\n');
  end
  fclose(fid);
```

谢 谢！