

年级、专业\_\_\_\_\_ 姓名\_\_\_\_\_ 学号\_\_\_\_\_ 名单序号（勿换行）\_\_\_\_\_

实验时间 2019 年 6 月 日

选课时间（打√）：周一 56; 周三 56 ☐ 周一 78; 周三 78 ☐

注：最后一部分是实验小结与收获，要求具体，详实，忌笼统

## 实验报告：实验 6 网站排名问题

1. 将附录A、B中的 $p=0.85$ 修改为 $p=0.80$ 、 $p=0.75$  或 $p=0.90$ 等以后，观察PageRank得分和排名结果的变化情况。

```
1 %fuluA
2 clear;
3 G=[ 0 0 0 1 0 1;
4     1 0 0 0 0 0;
5     0 1 0 0 0 0;
6     0 1 1 0 0 0;
7     0 0 1 0 0 0;
8     0 0 1 0 1 0;
9     ];
10 [n,n]=size(G);
11 sn=sum(G,1);
12 %Power Method
13 p=0.90;
14 % p=0.85;
15 % p=0.80;
16 % p=0.75;
17 delta=(1-p)/n;
18 D=zeros(n,1);
19 for j=1:n
20     if(sn(j)==0)
21         D(j)=1/n;
22         G(:,j)=ones(n,1);
23     end
24 end
25 p,G
26
```

```
1 p = 0.9000
2          按paeRank得分排名
3 排名(Rank)   PaeRank得分(x)   顶点(Node)   原始序号 (Index)
4 -----
5 1           0.269659         alpha         1
6 2           0.259308         beta          2
7 3           0.173327         delta         4
8 4           0.133312         gamma         3
9 5           0.107713         sigma         6
```

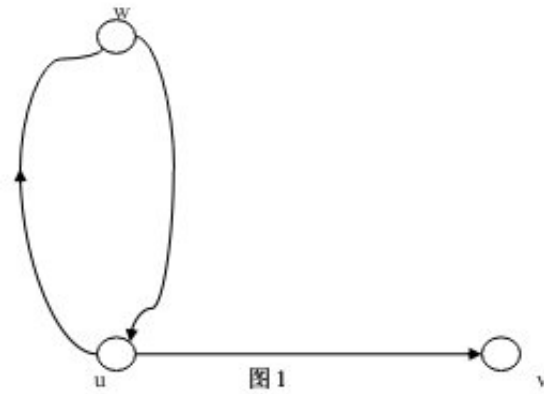
```

10 6          0.056681      rho          5
11 -----
12 p = 0.8500
13      按paeRank得分排名
14 排名(Rank)   PaeRank得分(x)   顶点(Node)   原始序号 (Index)
15 -----
16 1          0.267490      alpha        1
17 2          0.252418      beta         2
18 3          0.169769      delta        4
19 4          0.132302      gamma        3
20 5          0.115555      sigma        6
21 6          0.062467      rho          5
22 -----
23 p = 0.8000
24      按paeRank得分排名
25 排名(Rank)   PaeRank得分(x)   顶点(Node)   原始序号 (Index)
26 -----
27 1          0.265097      alpha        1
28 2          0.245356      beta         2
29 3          0.166538      delta        4
30 4          0.131463      gamma        3
31 5          0.123137      sigma        6
32 6          0.068409      rho          5
33 -----
34 p = 0.7500
35      按paeRank得分排名
36 排名(Rank)   PaeRank得分(x)   顶点(Node)   原始序号 (Index)
37 -----
38 1          0.262164      alpha        1
39 2          0.238356      beta         2
40 3          0.163795      delta        4
41 4          0.131053      gamma        3
42 5          0.130222      sigma        6
43 6          0.074409      rho          5
44 -----

```

不同的 P 值对排名并没有变化

2.为计算图1所示的小型网络的排名，请修改附录A、B的程序；



```

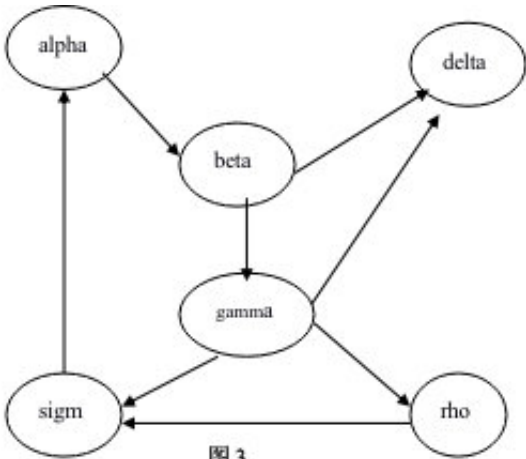
1  %fuluA&B
2  clear;
3  G=[0,1,0;
4      1,0,0;
5      1,0,0];
6  [n,n]=size(G);
7  sn=sum(G,1);
8  %Power Method
9  % p=0.90;
10 % p=0.85;
11 p=0.80;
12 % p=0.75;
13 delta=(1-p)/n;
14 D=zeros(n,1);
15 for j=1:n
16     if(sn(j)==0)
17         D(j)=1/n;
18         G(:,j)=ones(n,1);
19     else
20         D(j)=1/sn(j);
21     end
22 end
23 D=diag(D);
24 A=p*G*D+delta*ones(n);
25 x=ones(n,1)/n;
26 z=zeros(n,1);
27 cnt=0;
28 while max(abs(x-z))>0.0001
29     z=x;
30     x=A*x;
31     cnt=cnt+1;
32 end
33 [x1,index]=sort(x);
34 x1=flipud(x1);
35 p,x1,index

```

1	按paeRank得分排名			
2	排名(Rank)	PaeRank得分(x)	顶点(Node)	原始序号(Index)
3	-----			
4	1	0.391274	alpha	1
5	2	0.304363	gamma	3
6	3	0.304363	beta	2
7	-----			

将矩阵 G 改成[0,1,0;1,0,0;1,0,0;];选取 p=0.8 得到 U 的排名是第2,W 的排名是第 3,v 的排名是第 1.

3. 分析、改正附录C程序中的问题，并用于处理图3所示的小型网络的排名计算；



```

1 clear;
2 G=[ 0 0 0 0 0 1;
3     1 0 0 0 0 0;
4     0 1 0 0 0 0;
5     0 1 1 0 0 0;
6     0 0 1 0 0 0;
7     0 0 1 0 1 0;
8 ];
9 [n,n]=size(G);
10 sn=sum(G,1);
11 %Power Method
12 % p=0.90;
13 % p=0.85;
14 p=0.80;
15 % p=0.75;
16 delta=(1-p)/n;
17 D=zeros(n,1);
18 for j=1:n
19     if(sn(j)==0)
20         D(j)=1/n;
21         G(:,j)=ones(n,1);
22     else
```

```

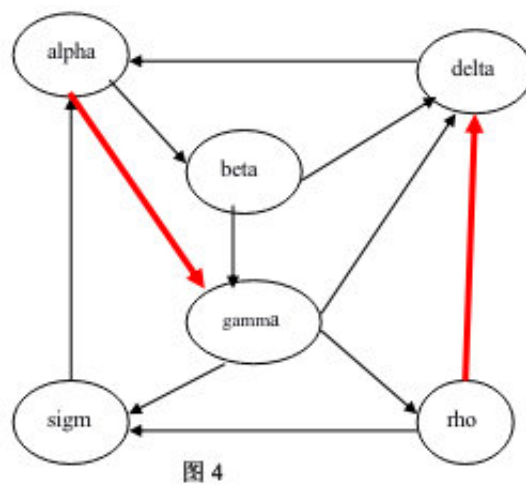
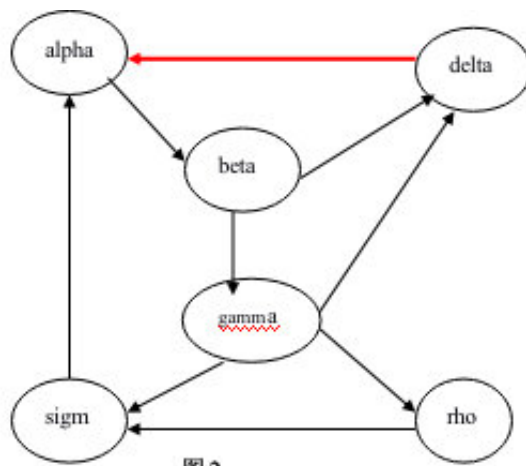
23     D(j)=1/sn(j);
24     end
25 end
26 D=diag(D);
27 A=p*G*D+delta*ones(n);
28 x=ones(n,1)/n;
29 z=zeros(n,1);
30 cnt=0;
31 while max(abs(x-z))>0.0001
32     z=x;
33     x=A*x;
34     cnt=cnt+1;
35 end
36
37 [x1,index]=sort(x);
38 x1=flipud(x1);
39 index=flipud(index);
40 title={'排名(Rank)','PaeRank得分(x)','顶点(Node)','原始序号 (Index)'};
41 sites={"alpha","beta","gamma","delta","rho","sigma"};
42 fprintf("                按paeRank得分排名\n");
43 fprintf("");
44 fprintf("%-11s%-16s%-11s\n",title{1},title{2},title{3},title{4});
45 fprintf("-----\n");
46 for i=1:6
47     fprintf("%-11d%-16f%-11s d\n",i,x1(i),sites{index(i)},index(i));
48 end

```

	按paeRank得分排名			
	排名(Rank)	PaeRank得分(x)	顶点(Node)	原始序号 (Index)
3	-----			
4	1	0.213529	beta	2
5	2	0.195115	alpha	1
6	3	0.180935	delta	4
7	4	0.172025	sigma	6
8	5	0.142843	gamma	3
9	6	0.095554	rho	5

结合题目 1 的代码,改变邻接矩阵 G,取  $p=0.80$ ,得到 6 个网站的排名如上

4. 计算图4所示的小型网络的排名, 分析其排名与图2所示的小型网络的排名发生变化的原因;



在题目 3 的代码基础上改变邻接矩阵 G

```

1
2 %fuluC-4-1
3 clear;clc;
4 G=[0 0 0 1 0 1;
5     1 0 0 0 0 0;
6     0 1 0 0 0 0;
7     0 1 1 0 0 0;
8     0 0 1 0 0 0;
9     0 0 1 0 1 0;
10    ];
11 [n,n]=size(G);
12 sn=sum(G,1);
13 %Power Method
14 % p=0.90;
15 % p=0.85;
16 p=0.80;
17 % p=0.75;
18 delta=(1-p)/n;
19 D=zeros(n,1);
20 for j=1:n

```

```

21     if(sn(j)==0)
22         D(j)=1/n;
23         G(:,j)=ones(n,1);
24     else
25         D(j)=1/sn(j);
26     end
27 end
28 D=diag(D);
29 A=p*G*D+delta*ones(n);
30 x=ones(n,1)/n;
31 z=zeros(n,1);
32 cnt=0;
33 while max(abs(x-z))>0.0001
34     z=x;
35     x=A*x;
36     cnt=cnt+1;
37 end
38
39 [x1,index]=sort(x);
40 x1=flipud(x1);
41 index=flipud(index);
42 title={'排名(Rank)', 'PaeRank得分(x)', '顶点(Node)', '原始序号 (Index)'};
43 sites={"alpha", "beta", "gamma", "delta", "rho", "sigma"};
44 fprintf("                按paeRank得分排名\n");
45 fprintf("");
46 fprintf("%-11s%-16s%-11s\n", title{1}, title{2}, title{3}, title{4});
47 fprintf("-----\n");
48 for i=1:6
49     fprintf("%-11d%-16f%-11s d\n", i, x1(i), sites{index(i)}, index(i));
50 end
51 %fuluC-4-1
52 clear;
53 G=[0 0 0 1 0 1;
54     1 0 0 0 0 0;
55     1 1 0 0 0 0;
56     0 1 1 0 1 0;
57     0 0 1 0 0 0;
58     0 0 1 0 1 0;
59     ];
60 [n,n]=size(G);
61 sn=sum(G,1);
62 %Power Method
63 % p=0.90;
64 % p=0.85;
65 p=0.80;
66 % p=0.75;
67 delta=(1-p)/n;
68 D=zeros(n,1);
69 for j=1:n

```

```

70     if(sn(j)==0)
71         D(j)=1/n;
72         G(:,j)=ones(n,1);
73     else
74         D(j)=1/sn(j);
75     end
76 end
77 D=diag(D);
78 A=p*G*D+delta*ones(n);
79 x=ones(n,1)/n;
80 z=zeros(n,1);
81 cnt=0;
82 while max(abs(x-z))>0.0001
83     z=x;
84     x=A*x;
85     cnt=cnt+1;
86 end
87
88 [x1,index]=sort(x);
89 x1=flipud(x1);
90 index=flipud(index);
91 title={'排名(Rank)','PaeRank得分(x)','顶点(Node)','原始序号 (Index)'};
92 sites={"alpha","beta","gamma","delta","rho","sigma"};
93 fprintf("                按paeRank得分排名\n");
94 fprintf("");
95 fprintf("%-11s%-16s%-11s%\n",title{1},title{2},title{3},title{4});
96 fprintf("-----\n");
97 for i=1:6
98     fprintf("%-11d%-16f%-11s% d\n",i,x1(i),sites{index(i)},index(i));
99 end

```

按paeRank得分排名				
排名(Rank)	PaeRank得分(x)	顶点(Node)	原始序号 (Index)	
-----				
1	0.265097	alpha	1	
2	0.245356	beta	2	
3	0.166538	delta	4	
4	0.131463	gamma	3	
5	0.123137	sigma	6	
6	0.068409	rho	5	

按paeRank得分排名				
排名(Rank)	PaeRank得分(x)	顶点(Node)	原始序号 (Index)	
-----				
1	0.272584	alpha	1	
2	0.199370	gamma	3	
3	0.178065	delta	4	



17	4	0.142393	beta	2
18	5	0.121088	sigma	6
19	6	0.086500	rho	5

## 5. 查看工具箱ncm中的文件surfer.m的作用；

```

1 function [U,G] = surfer(root,n)
2 % SURFER Create the adjacency graph of a portion of the Web.
3 % [U,G] = surfer(root,n) starts at the URL root and follows
4 % Web links until it forms an adjacency graph with n nodes.
5 % U = a cell array of n strings, the URLs of the nodes.
6 % G = an n-by-n sparse matrix with G(i,j)=1 if node j is linked to
   node i.
7 %
8 % Example: [U,G] = surfer('http://www.harvard.edu',500);
9 % See also PAGERANK.
10 %
11 % This function currently has two defects. (1) The algorithm for
12 % finding links is naive. We just look for the string 'http:'.
13 % (2) An attempt to read from a URL that is accessible, but very
   slow,
14 % might take an unacceptably long time to complete. In some cases,
15 % it may be necessary to have the operating system terminate
   MATLAB.
16 % Key words from such URLs can be added to the skip list in
   surfer.m.
17
18 % Initialize
19 %SURFER创建Web的一部分的邻接图。
20 %[U, G] =冲浪者 (root, n) 从URL根开始, 然后是
21 %Web链接, 直到它形成一个包含n个节点的邻接图。
22 %U = n个字符串的单元格数组, 即节点的URL。
23 %G = n-by-n稀疏矩阵, 如果节点j链接到节点i, 则G(i, j) = 1。
24 %
25 %示例: [U, G] =surfer ('http: //www.harvard.edu',500) ;
26 %另见PAGERANK。
27 %
28 %此功能目前有两个缺陷。 (1) 算法
29 %发现链接是天真的。我们只是寻找字符串'http: '。
30 % (2) 尝试从可访问但非常慢的URL读取
31 %可能需要花费不可思议的长时间才能完成。在某些情况下,
32 %可能需要让操作系统终止MATLAB。
33 %这些URL中的关键词可以添加到surfer.m的跳过列表中。
34 %初始化

```

---

## 6. .用命令

`[U,G]=surfer('http://www.ouc.edu.cn',100)`

建立与中国海洋大学主页有关的100个网页的网址矩阵U和邻接矩阵G，并回答如下问题：

请问中国海洋大学图书馆主页排在第几名？PageRank是多少？它有多少个入度和出度？

**18** `http://library.ouc.edu.cn`

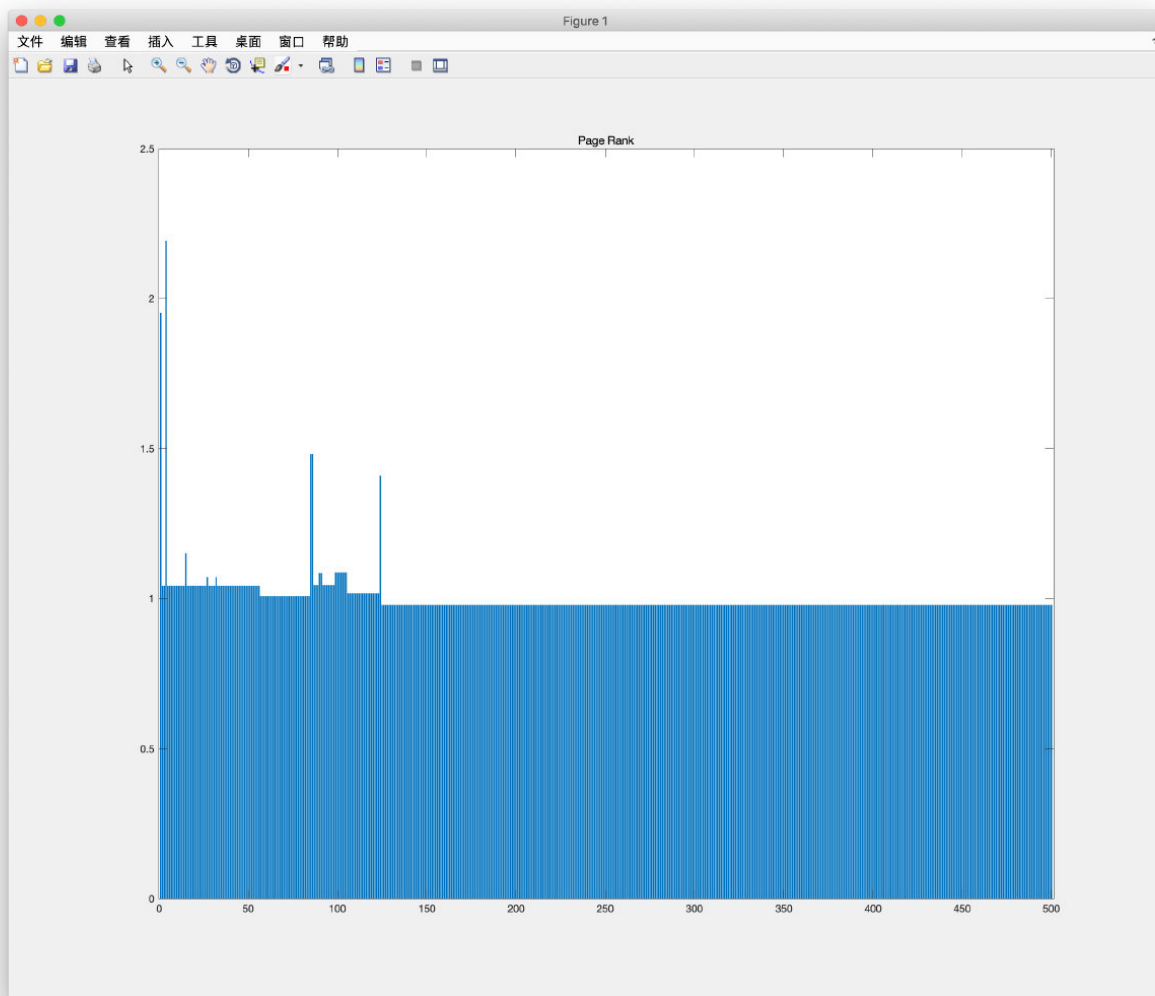
排名 18,出度为10,入度为1

---

## 7. 用命令

```
1 [U,G] = surfer('http://www.ouc.edu.cn',500)
2
3 [x,cnt]=pagerankpow(G)
4
5 pagerank(U,G)
```

建立与中国海洋大学主页有关的500个网页的网址矩阵 U和邻接矩阵G. 中国海洋大学数学数学院主页处于什么位置？



大约在 161 的位置,得分为 0.001315

实验中我开启是 shadowsock,保证了境外连接可以访问,所以结果会更加类似于 google .