

```

% 附录 A （例 4 的程序：fuluA.m）
clear
G=[...
0   0   0   1   0   1
1   0   0   0   0   0
0   1   0   0   0   0
0   1   1   0   0   0
0   0   1   0   0   0
0   0   1   0   1   0];
[n,n]=size(G);
sn=sum(G,1);    %% G 的列和
% Power Method
p=0.85;
delta=(1-p)/n;
D=zeros(n,1);
for j=1:n
    if sn(j)==0
        D(j)=1/n;
        G(:,j)=ones(n,1);
    else
        D(j)=1/sn(j);
    end
end
D=diag(D);
A=p*G*D+delta*ones(n);
x=ones(n,1)/n;
z=zeros(n,1);
cnt=0;
while max(abs(x-z))>0.0001
    z=x;
    x=A*x;
    cnt=cnt+1;
end
[x1,index]=sort(x);
x1=flipud(x1);
index=flipud(index);
title={'排名（Rank）','PageRank 得分(x)','顶点(Node)','原始序号(Index)'};
sites={'alpha','beta','gamma','delta','rho','sigma'};
fprintf('                按 PageRank 得分排名\n');
fprintf('-----');
fprintf('%-11s %-16s %-11s %s\n',title{1},title{2},title{3},title{4});
fprintf('-----\n');
for i=1:6
    fprintf('%-11d %-16f %-11s %d\n',i,x1(i),sites{index(i)},index(i));

```

```

end
%*****
% 附录 B （例 5 的程序：fuluB.m）
clear
G=[...
0   0   0   1   0   1
1   0   0   0   0   0
0   1   0   0   0   0
0   1   1   0   0   0
0   0   1   0   0   0
0   0   1   0   1   0]; % Link Structure
[n,n]=size(G);
p=0.85;
delta=(1-p)/n;
sn=sum(G,1); %按列求矩阵 G 各列的和
D=diag(1./sn);
A=p*G*D+delta;
% 幂迭代法
x=ones(n,1);%迭代初始向量
z=zeros(n,1);
cnt=0;% 用于记录迭代步数
while max(abs(x-z))>0.0001
    z=x;
    x=A*x;
    cnt=cnt+1;
end
[x1,index]=sort(x);
x1=flipud(x1);
index=flipud(index);
% 输出结果
out=[1:n;x1';index'];
fprintf('迭代步数=%d\n',cnt);
str1='排名';str2='PageRank 得分';str3='序号';
fprintf('%-6s %-15s %-5s\n',str1,str2,str3);
fprintf('%-6d %-15f %-5d\n',out);
%*****
% 附录 C （fuluC.m）
G=[...
0   0   0   1   0   1
1   0   0   0   0   0
0   1   0   0   0   0
0   1   1   0   0   0
0   0   1   0   0   0
0   0   1   0   1   0]; % Link Structure

```

```

[n,n]=size(G);
sn=sum(G,1); %按列求矩阵 G 各列的和
% Power Method
p=0.85;
delta=(1-p)/n;
D=zeros(n,1);
for j=1:n
    if sn(j)==0
        D(j)=1/n;
    else
        D(j)=1/sn(j);
    end
end
D=diag(D);
A=p*G*D+delta*ones(n);
x=ones(n,1)/n;
z=zeros(n,1);
cnt=0;
while max(abs(x-z))>0.0001
    z=x;
    x=A*x;
    cnt=cnt+1;
end
[x1,index]=sort(x);
x1=flipud(x1);
index=flipud(index);
% 输出结果
out=[1:n;x1';index'];
fprintf('迭代步数=%d\n',cnt);
str={'排名','PageRank 得分','序号'};
fprintf('%-6s %-15s %-5s\n',str{1},str{2},str{3});
fprintf('%-6d %-15f %-5d\n',out);
fprintf('-----按序号排列-----\n')
str={'序号','PageRank 得分','排名'};
x2=x; [tmp,rank]=sort(index); out=[1:n;x2';rank'];
fprintf('%-6s %-15s %-5s\n',str{1},str{2},str{3});
fprintf('%-6d %-15f %-5d\n',out);
%*****
% 附录 D （fuluD.m）
aa=cputime;
G=[...
0   0   0   1   0   1
1   0   0   0   0   0
0   1   0   0   0   0

```

```

0   1   1   0   0   0
0   0   1   0   0   0
0   0   1   0   1   0];
G=double(full(G));
% Link Structure
[n,n]=size(G);
for j=1:n
    L{j}=find(G(:,j));
    c(j)=length(L{j});
end
% Power Method
p=0.85;
delta=(1-p)/n;
x=ones(n,1)/n;
z=zeros(n,1);
cnt=0;
while max(abs(x-z))>0.0001
    z=x;
    x=ones(n,1);
    for j=1:n
        if c(j)==0
            x=x+z(j)/n;
        else
            x(L{j})=x(L{j})+z(j)/c(j);
        end
    end
    p=p*x+delta;
    cnt=cnt+1;
end
bb=cputime;
[x1,index]=sort(x);
x1=flipud(x1);
index=flipud(index);
% 输出结果
title={'排名 (Rank)', 'PageRank 得分(x)', '顶点(Node)', '原始序号(Index)'};
sites={'alpha', 'beta', 'gamma', 'delta', 'rho', 'sigma'};
fprintf('                                按 PageRank 得分排名\n');
fprintf('-----');
fprintf('%-11s %-16s %-11s %s\n', title{1}, title{2}, title{3}, title{4});
fprintf('-----\n');
for i=1:6
    fprintf('%-11d %-16f %-11s %d\n', i, x1(i), sites{index{i}}, index(i));
end
fprintf('程序运行时间为: %12.10g 秒\n', bb-aa);

```