```
%use the three methods with at least two initial vectors randomly generated
%by randn(n,1) to solve the following problems.  To ensure two generated
%vectors are different, you may consider setting the random number
%generator rng.  Compare their performance in terms of the number of
%iterations and the error (use l-inf-norm) between the actual eigenvalue/eigenvectors and
%their respective estimations.  If the initial random vectors are changed,
%then iter and error will be changed accordingly in the results.

% a) Find the rank of each webpage in the network showin in Figure 1 with 1
% webpages.  Construct adjacency matrix B an dthe modified adjacency matrix
% M, then execute eigfinder.m.


B=[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
  1 0 0 0 0 0 0 0 0 0 1 0 0 0 0
  0 1 0 0 1 0 0 0 0 0 0 0 0 0 0
  1 1 0 0 0 0 0 0 0 0 0 0 1 0 0
  0 0 0 0 0 1 0 1 0 0 0 0 0 0 0
  0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
  0 0 0 0 0 0 0 0 1 1 0 0 0 0 0
  0 0 0 0 1 0 0 0 0 0 1 0 0 0 0
  0 0 0 0 0 0 0 0 1 0 1 0 0 1 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
  0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
  0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
  0 0 0 0 0 0 0 0 0 0 0 0 0 1 0];

M=zeros(15);  %preallocate for speed
for i=1:15
        for j=1:15
        M(i,j)=.85*(B(i,j)/sum(B(i,:)))+(1-.85)/15;
        end
end

rng;  %set random number generator
x0=rand(15,1);

paras.tol=10^-6;
paras.maxiter=100;
paras.q=(x0'*M*x0)/(x0'*x0);
paras.option='power1';
```

```
[ lambda1, v1, iter1 ] = eigfinder( M, x0, paras );
 v1=v1/norm(v1);

paras.option='power2';
[ lambda2, v2, iter2 ] = eigfinder( M, x0, paras );
 v2=v2/norm(v2);

paras.option='invpower';
[ lambda3, v3, iter3 ] = eigfinder( M, x0, paras );
 v3=v3/norm(v3);

x0=rand(15,1);

paras.tol=10^-6;
paras.maxiter=100;
paras.q=(x0'*M*x0)/(x0'*x0);
paras.option='power1';

[ lambda4, v4, iter4 ] = eigfinder( M, x0, paras );
 v4=v4/norm(v4);


paras.option='power2';
[ lambda5, v5, iter5 ] = eigfinder( M, x0, paras );
 v5=v5/norm(v5);


paras.option='invpower';
[ lambda6, v6, iter6 ] = eigfinder( M, x0, paras );
 v6=v6/norm(v6);




%Compute the actual dominant eigenvalue and its associated eigenvector
[x,y]=eig(M);
eigvec_a=x(:,1);  %only look at the first column because that's the dominant
  %eigvalue/eigenvector
eigvalue_a=y(1,1);


error1=norm(abs(eigvec_a-v1),Inf)
error2=norm(abs(eigvec_a-v2),Inf)
error3=norm(abs(eigvec_a-v3),Inf)
```

```
error4=norm(abs(eigvec_a-v4),Inf)
error5=norm(abs(eigvec_a-v5),Inf)
error6=norm(abs(eigvec_a-v6),Inf)




% b) Find the dominant eigenvalue and the dominant eigenvector of the
% matrix A

rng;  %set random number generator

A=[2.395798 0.234169 0.127074 0.146184 0.183889
	0.113724 5.103374 0.243386 0.030779 0.241161
	0.183743 0.199444 7.642053 0.199313 0.145211
	0.085881 0.144653 0.104811 9.013056 0.024832
	0.053909 0.180566 0.126246 0.249744 3.774798];

x0=rand(5,1);

paras.tol=10^-6;
paras.maxiter=100;
paras.q=(x0'*A*x0)/(x0'*x0);

paras.option='power1';
[ lambda1b, v1b, iter1b ] = eigfinder( A, x0, paras );
 v1b=v1b/norm(v1b);


paras.option='power2';
[ lambda2b, v2b, iter2b ] = eigfinder( A, x0, paras );
 v2b=v2b/norm(v2b);


paras.option='invpower';
[ lambda3b, v3b, iter3b ] = eigfinder( A, x0, paras );
 v3b=v3b/norm(v3b);



x0=rand(5,1);

paras.tol=10^-6;
```

```
paras.maxiter=100;
paras.q=(x0'*A*x0)/(x0'*x0);
paras.option='power1';

[ lambda5b, v4b, iter4b ] = eigfinder( A, x0, paras );
 v4b=v4b/norm(v4b);


paras.option='power2';
[ lambda5b, v5b, iter5b ] = eigfinder( A, x0, paras );
 v5b=v5b/norm(v5b);


paras.option='invpower';
[ lambda6b, v6b, iter6b ] = eigfinder( A, x0, paras );
 v6b=v6b/norm(v6b);


%Compute the actual dominant eigenvalue and its associated eigenvector
[x,y]=eig(A);
eigvec_b=x(:,1);  %only look at the first column because that's the dominant
  %eigvalue/eigenvector
eigvalue_b=y(1,1);



error1b=norm(abs(-1.*eigvec_b-v1b),Inf)
error2b=norm(abs(-1.*eigvec_b-v2b),Inf)
error3b=norm(abs(-1.*eigvec_b-v3b),Inf)
error4b=norm(abs(-1.*eigvec_b-v4b),Inf)
error5b=norm(abs(-1.*eigvec_b-v5b),Inf)
error6b=norm(abs(-1.*eigvec_b-v6b),Inf)
```