Apellidos, Nombre: Cristóbal Pascual, David
Apellidos, Nombre: Doncel Aparicio, Alberto

1. A. Calcular la matriz G de pagerank del grafo

Código y volcado de las matrices/vectores/índices: C, Nj, dj, S, G

```
i=[1 4 7 2 1 3 6 7 1 6 3 2 4 6];
j=[2 2 2 3 4 4 4 4 5 5 6 7 7 7];
%i=find(Nj==0);
N=7; % DimensiÃ³n de la matriz
alpha=0.85;
C=sparse(j,i,1,N,N);
%C=sparse(1:7,1:7,1,N,N)
 % Crea la matriz dispersa C de tamaÃ±o NxN tal que C(j(k),i(k))=1.
 % Los vectores i, j del mismo tamaÃ±o contienen los nodos de entrada y de salida
full(C) % Visualizamos la matriz completa.
Ccompleta=full(C); % Creamos la matriz completa
whos % Vemos el tamaÃ±o que ocupan en memoria las matrices C y Ccompleta

Nj=sum(C);
Dj=zeros(1,N);
Dj(find(Nj==0))=1;
S=C;
for k=1:N
    if Dj(k)==1
        S(:,k)=ones(N,1)/N;
    else
        S(:,k)=S(:,k)/Nj(k);
    end
end
G=alpha*S+(1-alpha)*ones(N)/N;
[V D]=eig(G);
autovalores=diag(abs(D));
autovectores=sum(abs(V));

[lambda,x]=potencia(G,50);
precision1=norm(G*x-lambda*x);
precision2=abs(lambda-1);
```

```
C=0     0     0     0     0     0     0
     1     0     0     1     0     0     1
     0     1     0     0     0     0     0
     1     0     1     0     0     1     1
     1     0     0     0     0     1     0
     0     0     1     0     0     0     0
     0     1     0     1     0     1     0

Nj=3     2     2     2     0     3     2

Dj=[0,0,0,0,1,0,0]

S=      0         0         0         0    0.1429         0         0
    0.3333         0         0    0.5000    0.1429         0    0.5000
```

```
     0    0.5000         0         0    0.1429         0         0
0.3333         0    0.5000         0    0.1429    0.3333    0.5000
0.3333         0         0         0    0.1429    0.3333         0
     0         0    0.5000         0    0.1429         0         0
     0    0.5000         0    0.5000    0.1429    0.3333         0
```

S es estocástica (todas sus columnas suman 1)

```
G=    0.0214    0.0214    0.0214    0.0214    0.1429    0.0214    0.0214
    0.3048    0.0214    0.0214    0.4464    0.1429    0.0214    0.4464
    0.0214    0.4464    0.0214    0.0214    0.1429    0.0214    0.0214
    0.3048    0.0214    0.4464    0.0214    0.1429    0.3048    0.4464
    0.3048    0.0214    0.0214    0.0214    0.1429    0.3048    0.0214
    0.0214    0.0214    0.4464    0.0214    0.1429    0.0214    0.0214
    0.0214    0.4464    0.0214    0.4464    0.1429    0.3048    0.0214
```

1. B. Teorema de Perron-Frobenius

Dar comando de Matlab, visualizar y comentar el resultado y contestar a las preguntas. Volcar el contenido de la pantalla. No es necesario codificar ninguna rutina/código.

Teorema de Perron-Frobenius: Si S es una matriz irreducible, estocástica y no negativa entonces el mayor autovalor de S es 1 y el correspondiente autovector r tiene todos sus elementos no negativos. Al autovalor 1 y autovector r se les denomina autovalor y autovector dominantes.

Como podemos ver en lo impreso en el apartado anterior, G es estocástica (ya que todas sus columnas suman 1), es irreducible y es ninguno de sus elementos es negativo. Por lo tanto G cumple el teorema de Perron-Frobenius.

```
[V D]=eig(G);
autovalores=diag(abs(D));
autovectores=sum(abs(V));

autovalores =
    1.0000
    0.2715
    0.3623
    0.3623
    0.1331
    0.4250
    0.3712
mayor autovalor = 1
autovector asociado =
0.0658384759109582
0.536141308256703
0.293698531920058
0.501742889391490
0.138513143812541
0.190660351976983
0.560959692971585
```

Cumple el teorema ya que tiene un único autovector r>= 0 asociado al autovalor 1. Además 1 es el mayor autovalor en módulo (dominante)

1. C. Método de la potencia

Código de la rutina. Volcado de la ejecución.

```
function [autovalor, autovector]=potencia(A,niter)
   N=7;
   x1=ones(N,1);
   respuesta=x1;
   for k=1:niter
      x=x1;
      x=x/norm(x);
      x1=A*x;
   end
   lambda=x'*x1;
   autovector=x;
   autovalor=lambda;


return
```

Comandos, volcar el resultado y contestar a las preguntas.

```
[lambda,x]=potencia(G,50);
```

¿Son lambda/x los autovalor/autovector dominantes de la matriz G obtenidos antes con el comando eig?.

Sí.

```
lambda =

    1

x =

   0.0658
   0.5361
   0.2937
   0.5017
   0.1385
   0.1907
   0.5610
```

Utilizar la definición para comprobar que lambda y x son autovalor y autovector asociado de la matriz G.

Definición Gx = λx

```
G*x =     0.0658
   0.5361
   0.2937
   0.5017
   0.1385
   0.1907
   0.5610
λ*x =     0.0658
   0.5361
   0.2937
   0.5017
```

```
    0.1385
    0.1907
    0.5610
```

¿Todas las componentes del autovector dominante son del mismo signo?

Sí.

Calcular las precisiones numéricas obtenidas con el método de la potencia, aplicado a una matriz G que verifica el teorema de Perron-Frobenius:

```
precision1=norm(G*x-lambda*x);
precision2=abs(lambda-1);
Precision1=1.249000902703301e-16
Precision2=0
```

1. D. Calcular el pagerank del grafo

Código de la rutina.

```
function [autovalor, pagerank]=getPageRank(A,niter,N) %modificada para apartados 2 y 3
    x1=ones(N,1);
    for k=1:niter
        x=x1;
        x=x/norm(x);
        x1=A*x;
    end
    lambda=x'*x1;
    pagerank=x/sum(x);
    autovalor=lambda;
return
```

Volcado de la ejecución.

```
[lambda,pagerank]=getPageRank(G,50,7)

lambda =

     1

pagerank =

    0.0288
    0.2344
    0.1284
    0.2193
    0.0606
    0.0833
    0.2452
```

Contestar a las preguntas.

Comprobar que el vector pagerank es el pagerank de G. Dar los comandos necesarios para comprobar que el vector pagerank es autovector asociado al autovalor 1, todas sus componentes son positivas y suman 1.

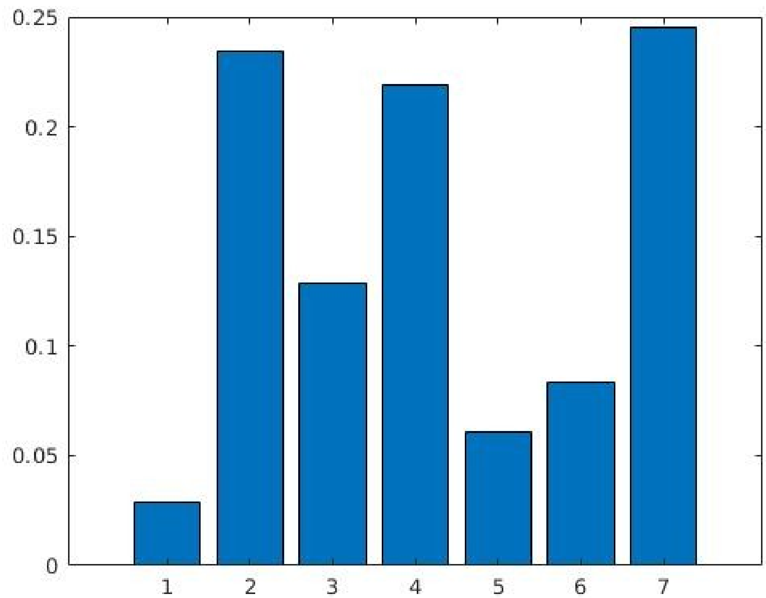
Como podemos comprobar por lo impreso justo encima todas sus componentes son positivas, pagerank esta asociado al autovalor 1 y sus componentes suman 1 como vemos aquí.

sum(pagerank)

ans =

1

Dar las dos precisiones obtenidas.
Precisión1= 1.249000902703301e
Precisión2= 0
Pegar la gráfica del comando bar.



Orden del pagerank del grafo.

P7>P2>P4>P3>P6>P5>P1

2.

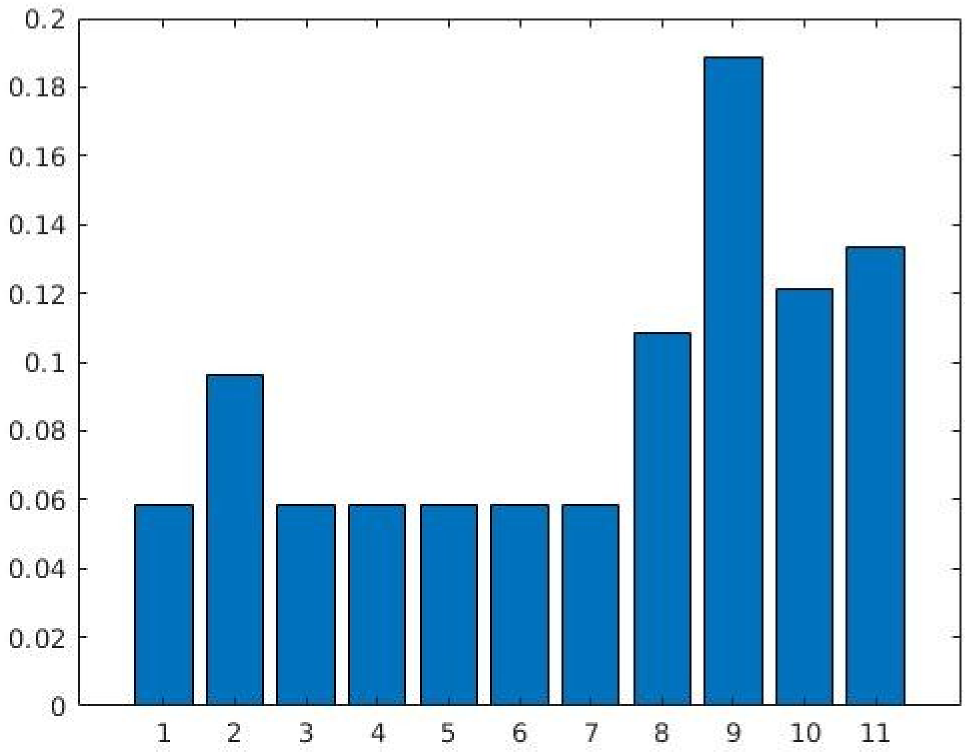
Vectores i=[5 3 3 7 7 8 11 11 11]
j=[11 10 8 8 11 9 9 10 2]

Script

```
i=[5 3 3 7 7 8 11 11 11];
j=[11 10 8 8 11 9 9 10 2];
N=11; % DimensiÃ³n de la matriz
alpha=0.85;
C=sparse(j,i,1,N,N);

Nj=sum(C);
Dj=zeros(1,N);
Dj(find(Nj==0))=1;
```

```
S=C;

for k=1:N
    if Dj(k)==1
        S(:,k)=ones(N,1)/N;
    else
        S(:,k)=S(:,k)/Nj(k);
    end
end

G=alpha*S+(1-alpha)*ones(N)/N;
[autovalor,pagerank]=getPageRank(G,50,N);
bar(pagerank)

function [autovalor, autovector]=potencia(A,niter)
    N=11;
    x1=ones(N,1);
    for k=1:niter
        x=x1;
        x=x/norm(x);
        x1=A*x;
    end
    lambda=x'*x1;
    autovector=x;
    autovalor=lambda;
return
end

function [autovalor, pagerank]=getPageRank(A,niter,N)
    x1=ones(N,1);
    for k=1:niter
        x=x1;
        x=x/norm(x);
        x1=A*x;
    end
    lambda=x'*x1;
    pagerank=x/sum(x);
    autovalor=lambda;
return
end
```

Gráfica del comando bar(pagerank)



Orden de las páginas.
P9>P11>P10>P8>P2>P1=P3=P4=P5=P6=P7

3.

Script

```
N=10;p=5;i=randi(N,1,p*N);j=randi(N,1,p*N);C=sparse(i,j,1,N,N);niter=20;
%N=dimensiÃ³n, p=#medio links salida, niter=# iteraciones
alpha=0.85;
Nj=sum(C);
Dj=zeros(1,N);
Dj(find(Nj==0))=1;
S=C;

for k=1:N
    if Dj(k)==1
        S(:,k)=ones(N,1)/N;
    else
        S(:,k)=S(:,k)/Nj(k);
    end
end

G=alpha*S+(1-alpha)*ones(N)/N;
[lambda,x]=potencia(G,50);
precision1=norm(G*x-lambda*x);
precision2=abs(lambda-1);
[lambda,pagerank]=getPageRank(G,50,N);
```

```
bar(pagerank);
precision=max(precision1,precision2)
ordenpagerank=sort(pagerank,'descend')

function [autovalor, pagerank]=getPageRank(A,niter,N)
   x1=ones(N,1);
   for k=1:niter
      x=x1;
      x=x/norm(x);
      x1=A*x;
   end
   lambda=x'*x1;
   pagerank=x/sum(x);
   autovalor=lambda;
return
end

function [autovalor, autovector]=potencia(A,niter)
   N=10;
   x1=ones(N,1);
   for k=1:niter
      x=x1;
      x=x/norm(x);
      x1=A*x;
   end
   lambda=x'*x1;
   autovector=x;
   autovalor=lambda;
return
end
```

vector ordenpagerank =

```
0.156764227297230
0.152178668398703
0.139948284953753
0.133148121861801
0.0996254468859387
0.0802864322000103
0.0790001242605280
0.0745901850098360
0.0588915662558061
0.0255669428763934

precision=1.415262216750919e-16
```