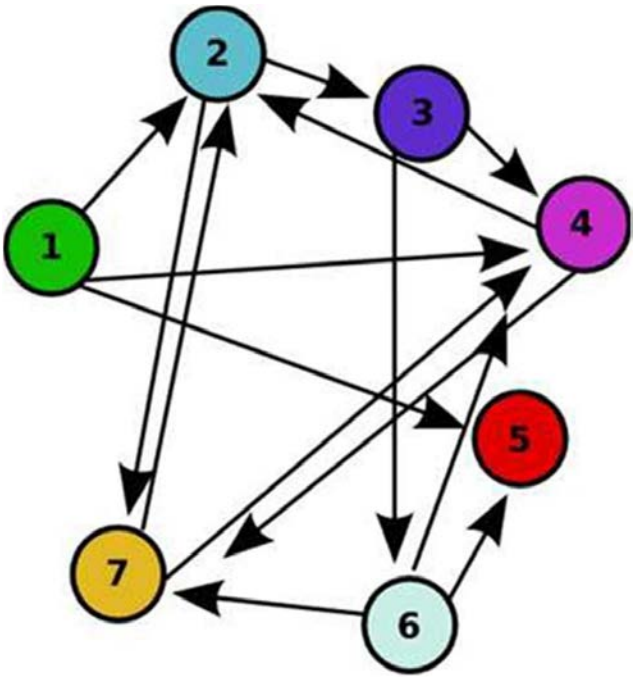
Calcular e interpretar el pagerank de un grafo

1. Sea el siguiente grafo 7x7.



Se desea calcular e interpretar su pagerank, utilizando la nomenclatura y definiciones de las presentaciones del curso.

Instrucciones.

- La codificación deberá ser lo más eficaz, eficiente y precisa posible y fácilmente escalable (pensar que lo que codifiquéis para una matriz de tamaño académico, lo tendréis que reescalar a una dimensión muy grande).
- Tener en cuenta las prestaciones numéricas del código (recursos que utiliza, memoria, cpu…).
- Se recomienda utilizar indexación lógica.
- En cada apartado se deberá entregar: el código utilizado, volcado de los datos, las gráficas, contestar a las preguntas e incluir comentarios, según corresponda en cada caso.

Comandos de Matlab:

- ***sparse*** define una matriz dispersa (son aquellas que tienen muy pocos elementos distintos de cero) y utiliza muy poco espacio en memoria (solo se almacena los índices y coeficientes no nulos).

- ***full*** para visualizar una matriz dispersa en formato completo, full(C). También convierte una matriz *sparse* en completa, B=full(C).

- ***whos*** para ver el tipo de matriz que estáis utilizando y su tamaño.

- Utilizaremos indexación lógica con el comando ***i=find(Nj==0)***; crea un vector de índices ***i*** para los cuales se verifica ***Nj(i)=0***.

- El comando ***[V D]=eig(A)***; calcula los autovalores y autovectores de la matriz A. Dar ***help eig*** para ver el formato. Los autovalores de A están en la diagonal de ***abs(D)***, los autovectores asociados están en las columnas de ***abs(D)***.

- ***randi*** para definir vectores de números enteros aleatorios. Lo utilizamos para definir matrices C de gran tamaño de una forma aleatoria.

- ***memory*** para ver el tamaño máximo de las tablas definidas en Matlab.
- ***tic*** al inicio de una función/script ***t=toc*** al final de la función/script para medir el tiempo de cpu utilizado.

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

- Definir la matriz de conectividad C del grafo 7x7 con el comando *sparse* de Matlab.

Con el siguiente código defines una matriz de conectividad, de tipo sparse, C con las 3 flechas que salen del nodo 1 y las 2 flechas que salen del nodo 2. Completar los vectores i,j para definir la matriz de conectividad C del grafo de la figura.

```
clear all
i=[1 1 1 2 2]; % Origen de las flechas: 3 links de salida del nodo 1 y 2 links de
salida del nodo 2
j=[2 4 5 3 7]; % Destino de las flechas: 3 links de entrada desde el nodo 1 y 2
links de entrada desde nodo 2
N=7; % Dimensión de la matriz
C=sparse(j,i,1,N,N) %Matriz C con 5 1s, C(i,j)=1 si Pj -> Pi
```

C=sparse(j,i,1,N,N);
% Crea **la matriz dispersa C** de tamaño NxN tal que C(j(k),i(k))=1.
% Los vectores i, j del mismo tamaño contienen los nodos de entrada y de salida

```
full(C)        %  Visualizamos la matriz completa.
Ccompleta=full(C)   % Creamos la matriz completa
whos          % Vemos el tamaño que ocupan en memoria las matrices C y Ccompleta
```

- Utilizar el comando *sum* para calcular el vector Nj que contiene el número de links de salida de cada nodo. ¿Tienen salida todos los nodos?. En su caso, ¿Qué nodos no tienen salida?. Comprobarlo en el grafo.

- A partir del vector Nj calcular el vector dj de los nodos sin salida:

$$dj(k)=\begin{cases}1 & \text{si } Nj(k)=0 \text{ (nodo sin salida)}\\ 0 & \text{si } Nj(k)\neq 0 \text{ (nodo con salida)}\end{cases}$$

Utilizar **indexación lógica.** % **i=find(Nj==0)**;

- A partir de la matriz C y el vector NJ, calcular la matriz S utilizando un bucle for (recorriendo las columnas).

  Comprobar que la matriz S es estocástica por columnas.

- Calcular la matriz G, con el parámetro $\alpha = 0.85$.

$$G=\alpha\, S+(1-\alpha)\, ones(N)/N$$

1. B. Teorema de Perron-Frobenius

- Comprobar si la matriz G verifica el teorema de Perron-Frobenius. Comprobar que verifica todas las condiciones de entrada y todas las condiciones de salida.

  - **Las condiciones de entrada. Enunciarlas y comprobar que G las verifica**.

    Podéis dar comandos de Matlab para visualizarlo por pantalla e interpretar los resultados, o indicar porqué se verifican las condiciones.

  - **Las condiciones de salida. Enunciarlas y comprobar que G las verifica.**

Utilizar el comando *eig(G)* para obtener los 7 autovalores de la matriz A y sus correspondientes autovectores. Indicar cuál es el mayor autovalor (en valor absoluto) y su autovector asociado. Indicar porqué se verifican las condiciones.

1. C. Método de la potencia para calcular el autovalor/autovector dominante de la matriz G

La solución de la ecuación $Ax = \lambda x$ se llama λ el **autovalor** y x el **autovector** de la matriz A.
Se llama autovalor y autovector dominantes cuando el autovalor es el de mayor módulo.

Si la matriz A verifica el teorema de Perron-Frobenius el siguiente método iterativo (método de la potencia, es similar a la tabla de estados de la presentación de clase) converge al autovalor y autovector dominantes de la matriz A.

- Codificar la siguiente rutina que implementa el método de la potencia.

[autovalor,autovector]=potencia(A,niter)

$$\begin{cases} x1=\text{ones(N,1)}; \ \% \text{ N dimensión del grafo y x1 vector arranque} \\ \text{for k= 1:niter} \quad \% \text{ niter número de iteraciones} \\ \quad x = x1; \quad x = x / norm(x); \\ \quad x1 = A * x; \\ \text{end} \\ \lambda = x' * x1; \ \% \text{ x' transpuesta de x} \end{cases}$$

Las variables de entrada son la matriz A y el número de iteraciones niter.
Las variables de salida son el autovalor dominante de A (lambda en el código) y su autovector asociado (x en el código).

- Ejecutar el método de la potencia para la matriz G y 50 iteraciones: [lambda,x]=potencia(G,50)

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

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

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

- Calcular las precisiones numéricas obtenidas con el método de la potencia:

$$\text{precision1} = \|Gx - x\|_2, \quad \text{precision2} = |\lambda - 1|$$
$$\text{precision} = \max(\text{precision1}, \text{precision2})$$

1. D. Calcular el *pagerank* del grafo

- Con el método de la potencia se obtiene (lambda,x) autovalor/autovector dominantes de la matriz G.
Comprobar que cualquier múltipmo del vector x también es autovector dominante de la matriz G.
Elegimos x=x/sum(x) que es un autovector dominante, con todas sus componentes del mismo signo y cuyas componentes suman 1 (es único). Le llamamos **pagerank** de la matriz G

- Modificar la rutina de la potencia para obtener como variable de salida el pagerank de la matriz G.

[lambda,pagerank]=potencia(G,niter)

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.

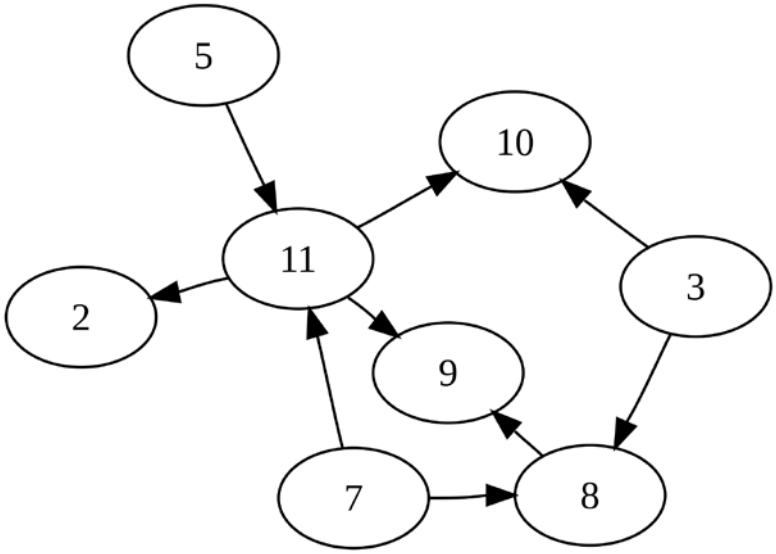
- Ejecutar la rutina de la potencia a la matriz G, con un niter suficiente grande para obtener el vector ***pagerank*** con al menos 12 cifras de precisión (precision1 y precisión2 <= 1e-12).
- Dar el comando ***bar(pagerank)*** para visualizar los resultados. ¿En qué orden mostrará Google las páginas del grafo?. Podéis utilizar la notación $P_1 > P_2 > P_3 > \ldots$

2. Scipt para calcular el pagerak de un grafo de tamaño 11x11.

Repetir el apartado anterior para calcular el pagerank del grafo de tamaño 11x11 de la figura.

Hacer un script para calcular el pagerank del grafo, de la forma más eficiente posible.
No es necesario comprobar el teorema de Perron-Frobenius.
El scrip debe empezar con la definición de los vectores i,j y la dimensión N.
Debe incluir la codificación mas eficiente y eficaz posible y terminar con el comando *bar(pagerank)* y ordenar las páginas según su pagerank.

Si el código del apartado anterior es el correcto, las modificaciones para hacer este apartado son mínimas.



3. Calculo de prestaciones numéricas. Scipt o función para calcular el pagerak de una matriz generada aleatoriamente.

Los siguientes comandos

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;

Siendo: N=dimensión, p=#medio links salida de cada página, niter=# iteraciones

Crean una matriz de conectividad C de dimensión 10x10 (los elementos pueden ser 1, 2, 3,…) de una forma aleatoria. Los elementos de C pueden ser 0, 1, 2, 3…. Consideramos a C como la matriz de conectividad.

Escribir el script o la función para calcular la matriz G y su pagerank.

[ordenpagerank,precision]=CalculoPageRank(N,p,niter)

El script tiene como variables de entrada: N (dimensión), p (número medio de links), niter (número de iteraciones).

El script genera la matriz C de forma aleatoria, calcula la matriz G y su pagerank.
La salida del script debe ser el vector ordenpagerank=1:N ordenado según el pagerank, y la precisión obtenida.

El script debe ser válido para cualquier valor de las variables de entrada.

Ejecutar el script para las variables de entrada:

- **N=1000, p= 20, niter=100.**
- **N=10000, p=20, niter=1000.**
- **N=100000, p=20,niter=1000.**
- **…**

Aumentar el tamaño de N hasta alcanzar la capacidad del sistema (tiempo cpu, memoria).
Hacer una tabla con los resultados.
Para cada valor de N indicar la memoria utilizada, el tiempo de cpu y la precisión obtenida.
Comentar las prestaciones numéricas del código CalculoPageRank.