**Aplicaciones numéricas de la Informática**

**Práctica 2: Calcular el pagerank de matrices de gran tamaño**

**Rubén Ibáñez Redondo**

**Mario Martín López**

1. Codificar la rutina para generar matrices aleatorias A dispersas de dimensión N

Rutina pedida:

function A=m\_aleat\_A(N,r)

% N=dim(A)

% r=número medio de links entrada/salida de cada nodo

p=randi(N,1,r\*N);

q=randi(N,1,r\*N);

A=sparse(p,q,1,N,N);

Nj=sum(A);

for k=1:N

if Nj(k) ~= 0

A(:,k)= A(:,k)/Nj(k);

end

end

return

Volcado de datos:

>> N=50; r=5; A=m\_aleat\_A(N,r); whos;

Name Size Bytes Class Attributes

A 50x50 4152 double sparse

N 1x1 8 double

ans 1x1 8 double

r 1x1 8 double

toolbox\_version 1x5 10 char

>> nnz(A)

ans =

234

Preguntas:

Comprobar:

- A es una matriz dispersa de tamaño NxN.

Sí, es cierto, el tamaño de A es 50x50, es decir, NxN

-Número de elementos no nulos de A. ¿Coincide con r\*N?.

234 elementos de A son no nulos, lo sabemos gracias al comando nnz(A)

El resultado de r\*N es 5\*50 = 250, casi igual que 234, pero en este caso diremos que sí coinciden debido a que el algoritmo no muestra enlaces de una misma página, por lo que muestra un número un poco menor a 250.

-El tamaño que ocupa en memoria.

A ocupa en memoria 4152 bytes

-La suma de cada columna de A es 1 o 0.

Mirando los valores de Nj, comprobamos que la suma las columnas es siempre 1 o 0.

Volcado de valores segundo caso:

>> N=1e+5;r=20;A=m\_aleat\_A(N,r);whos;nnz(A)

Name Size Bytes Class Attributes

A 100000x100000 32796840 double sparse

N 1x1 8 double

Nj 1x50 1192 double sparse

ans 1x1 8 double

r 1x1 8 double

toolbox\_version 1x5 10 char

ans =

1999802

Preguntas:

Comprobar:

- A es una matriz dispersa de tamaño NxN.

Sí, es cierto, el tamaño de A es 100000x100000, es decir, NxN

-Número de elementos no nulos de A. ¿Coincide con r\*N?.

1999802 elementos de A son no nulos, lo sabemos gracias al comando nnz(A)

El resultado de r\*N es 20\*100000 = 2000000, casi igual que 1999802, pero en este caso diremos que sí coinciden debido a que el algoritmo no muestra enlaces de una misma página, por lo que muestra un número un poco menor a 2000000.

-El tamaño que ocupa en memoria.

A ocupa en memoria 32796840 bytes

-La suma de cada columna de A es 1 o 0.

Mirando los valores de Nj, comprobamos que la suma las columnas es siempre 1 o 0.

-¿Hasta qué valor de N (tamaño de la matriz A), tienes memoria suficiente para definir la matriz A?.

Comprobamos que con N=1e+7 el ordenador no tiene suficiente memoria para definir la matriz A, asique el límite del valor de N será N=1e+6

1. Codificar una rutina para calcular el pagerank de una matriz G (densa) a partir de la matriz A (dispersa).

Rutina pedida:

function [x,precision1,t]=calculo\_PR(A,n\_iter)

N=length(A); % N es la dimensión de A

tic % Arranca el contador de tiempo

et=ones(N,1); % Vector columna de 1’s

alfa=0.85; % Parametro alfa

e=ones(1,N); % Vector fila de 1’s

Nj=sum(A); % Calcular Nj

j=[find(Nj==0)]; i=[1]; dj=sparse(i,j,1,1,N);% Calcular dj vector sparse

x1=ones(N,1);% Vector de arranque

v = alfa\*dj + (1-alfa)\*e % Calcular el vector v

% Empieza bucle:

for k= 1 : n\_iter

x = x1; x = x/norm(x);

x1= alfa\*A\*x + (1/N) \*et\*(v\*x) ;

end

% Fin del bucle

precision1=norm((alfa\*A\*x + (1/N) \*et\*(v\*x))-x1) % Calcular la precisión1=norm(G\*x-x)

x=x/sum(x); % Normalizar: x=x/sum(x);

t=toc % Para el contador de tiempo

return

>>N=1e+3,r=20,A=m\_aleat\_A(N,r);n\_iter=100;[x,precision1,t]=calculo\_PR(A,n\_iter);

N =

1000

r =

20

precision1 =

0

t =

0.0318