TEMA 6: **DISEÑO DE CLASES**

Práctica 10: **PageRank**

¿Te has planteado por qué al hacer una búsqueda en Google unas páginas aparecen antes que otras? Cada página tiene un ranking y cuanto más alto es, antes aparece en el listado final de la búsqueda. En esta práctica vas a simular un mundo en donde solo hay unas pocas páginas web y tendrás que calcular su ranking, también llamado *PageRank*.

Hallar el PageRank[1]

PageRank es un algoritmo (con copyright de Google) que mide la importancia de una página. Una página será más importante cuando cumpla que:

- Muchas páginas le hagan referencia. Si muchas páginas hacen referencia a una página *P*, significa que *P* es una *autoridad*.
- Si una o varias autoridades hacen referencia a otra página, quiere decir que esta última tiene cosas importantes que decir, y su *PageRank* aumenta.

Por ejemplo, la página web de Microsoft tiene un *PageRank* elevado ya que hay muchas páginas que la enlazan. Y si por un casual, creáis una página web que a Microsoft le parece interesante y mete en su página un link a la vuestra, tendríais un *PageRank* altísimo que os haría aparecer en las primeras posiciones de una búsqueda.

Para explicar cómo es el algoritmo de *PageRank*, supongamos que solo tenemos tres páginas web (N = 3), P_0, P_1 y P_2. Inicialmente, el *PageRank* es la probabilidad de que al abrir el navegador estemos en una de las páginas. Consideraremos que tenemos la misma probabilidad de empezar nuestra navegación en cualquiera de las páginas web de nuestro mundo. Por tanto, el *PageRank* inicial, PR_0, de cada página será $PR_0(P_0)=PR_0(P_1)=PR_0(P_2)=1/3$. PR_0 lo podemos expresar como un vector de columnas

$$PR_0=\begin{pmatrix}1/3\\1/3\\1/3\end{pmatrix}$$

El siguiente paso para hallar el *PageRank* de cada página es calcular la probabilidad de que tras un clic de ratón pasemos de una página web a otra. Supondremos que:

- Todos los links que hay en una página web tienen la misma probabilidad de ser pinchados. Por ejemplo, si la página P_0 tiene links a las páginas web P_1 y P_2, la probabilidad de visitar P_1 o P_2 desde P_0 es ½. Si la página web P_1 solo tiene un link a la página P_2, la probabilidad de visitar P_2 desde P_1 es 1, y la probabilidad de visitar P_0 desde P_1 es 0.
- Nuestra navegación no tiene que ser siempre pinchando en un link de la página que estamos visitando. Es habitual que, cansados de lo que estamos haciendo, de repente visitemos una página metiendo su dirección directamente en el navegador. De este modo, tenemos una probabilidad *d* (habitualmente se considera *d=0.85*) de seguir la navegación mediante los links y, por tanto, una probabilidad *1-d* (=0.15) de visitar otra página de nuestro mundo introduciendo la dirección en el navegador.

Todos estos datos los podemos meter en una matriz 3x3, *G*, donde las columnas expresan la probabilidad de visitar la página *i* (fila) desde la página *j* (columna)

[1] El algoritmo que se explica en esta práctica es la base del *PageRank* y es de dominio público. Sin embargo, Google guarda secretos de los detalles últimos de su algoritmo, de la "cocina" necesaria para preparar y explotar lo que se llama la matriz de Google.

	P_0	P_1	P_2
P_0	0*d + (1-d)/3	0*d + (1-d)/3	½*d + (1-d)/3
P_1	½*d + (1-d)/3	0 *d+ (1-d)/3	½*d + (1-d)/3
P_2	½ *d+ (1-d)/3	1*d + (1-d)/3	0 *d+ (1-d)/3

Con esta matriz podemos fácilmente hallar la probabilidad de estar en cualquier página tras un click de ratón, PR_1 (notar que *0.05 = (1-0.85)/3 0.475 = 1/2*0.85 + (1-0.85)/3) 0.9=1*0.85 + (1-0.85)/3*)

$$PR_1 = G \cdot PR_0 = \begin{pmatrix} 0.05 & 0.05 & 0.475 \\ 0.475 & 0.05 & 0.475 \\ 0.475 & 0.9 & 0.05 \end{pmatrix} \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix} = \begin{pmatrix} 0.575/3 \\ 1/3 \\ 1.425/3 \end{pmatrix}$$

Iterativamente, podemos calcular la probabilidad de estar en una página web tras n clicks de ratón mediante

$$PR_n = G \cdot PR_{n-1}$$

Se puede demostrar que si la matriz G (llamada ***Matriz de Google***) cumple ciertas condiciones de regularidad, y nuestras matrices las cumplirán, PR_n tiende a estabilizarse, esto es, a partir de un cierto n, la distancia entre PR_n y PR_{n+1} es muy pequeña , $\| PR_{n+1} - PR_n \| < \varepsilon$, y podemos suponer que $PR_n = PR_{n+1}$

El *PageRank* de cada página será el valor correspondiente del vector PR_n. Si te fijas, con el proceso iterativo anterior, hemos buscado un vector propio de la matriz G, ya que se cumple

$$PR_n = G \cdot PR_n$$

Curioso. Una de las bases del éxito de Google se basa en mezclar conocimientos de probabilidades, álgebra e informática. Para ordenar las páginas web de una búsqueda de Google, hay que hallar un vector propio de una matriz. En el ejemplo anterior, el *PageRank*[2] de cada página sería

$$PR(P_1) = 0.233, \quad PR(P_2) = 0.333, \quad PR(P_3) = 0.432$$

El proceso para hallar el *PageRank* de un mundo con N páginas web es:

1. Calcular la matriz de Google G: $l(p_i, p_j)$ es la probabilidad de llegar mediante un link a la página p_i desde p_j. Esta probabilidad es 0 si la página p_j no tiene ningún link a p_i y en caso de tenerlo, la probabilidad es igual a $1/N_j$, siendo N_j el número total de links que hay en la página p_j,

$$G = \begin{pmatrix} l(p_0, p_0)*d + (1-d)/N & \dots & l(p_0, p_0)*d + (1-d)/N \\ \dots & \dots & \dots \\ l(p_{N-1}, p_0)*d + (1-d)/N & \dots & l(p_{N-1}, p_{N-1})*d + (1-d)/N \end{pmatrix}$$

2. Empezar con el vector que indica la probabilidad de entrar en una página web al abrir el navegador

$$PR_0 = \begin{pmatrix} 1/N \\ 1/N \\ 1/N \end{pmatrix}$$

3. Iterativamente, calcular PR_n mediante

$$PR_n = G \cdot PR_{n-1}$$

hasta que para un cierto valor dado ε

$$\| PR_{n+1} - PR_n \| < \varepsilon$$

[2] El *PageRank* dado por Google es un número entero de 0 a 10.

Enunciado a resolver

Para realizar la simulación, vamos a suponer que tenemos 6 ficheros de texto, *t0.txt*,..., *t5.txt*, que representan páginas web con el siguiente formato:

- La primera palabra, y solo la primera palabra, que aparece en el fichero es el título de la página web.
- Si una página enlaza a otra, dentro del fichero se tendrá la palabra *link* seguida del nombre del fichero al que enlaza.

Ejemplo:

t0.txt

```
Vacas
Son animales link t1.txt
rumiantes con cuatro
estómagos link t2.txt
```

t1.txt

```
Animales
Ejemplos de animales son:
 rumiantes link t0.txt
```

.....

Los seis ficheros de texto necesarios para resolver el problema están creados y disponibles en el material adjunto a esta práctica.

Implementar un programa en Java para leer los seis ficheros y generar otro fichero de texto que contenga seis líneas, cada una con el título de una página web y su correspondiente ranking.

Observación:
El fichero de texto con la información de título y PageRank para el ejemplo propuesto debería contener:

```
Título: vacas. PageRank: 0.0999725401985748
Título: coches. PageRank: 0.19045140079213216
Título: casas. PageRank: 0.08944046710216397
Título: circos. PageRank: 0.18706136846152863
Título: principal. PageRank: 0.3297116509164225
Título: mariscos. PageRank: 0.10336257252917834
```