# II Proyecto Programado
# Mini-PageRank

UNIVERSIDAD NACIONAL
ESCUELA DE INFORMÁTICA
Estructuras de Datos

## 1. Introducción

El presente documento contiene la especificación del segundo proyecto programado de Estructuras de Datos para la Universidad Nacional.

## 2. Marco teórico

En el año 2000 aparece un buscador cuyo nombre puede ser familiar para muchos ahora, dado su éxito inicial, el nombre de este buscador es Google. Gran parte de su éxito se debió a que sus búsquedas eran muchos más precisas que sus otros (¿olvidados?) competidores.

El secreto para obtener búsquedas precisas consistió en el uso del page rank, el cual es un algoritmo basado en teoría de grafos, denominado Page Rank. En síntesis, representa la red como un grafo, donde cada nodo es una página web (url, al fin de cuentas) y cada enlace entre páginas es una arista, formando de esta manera un grafo dirigido[1]. Luego cuenta para cada nodo la cantidad de aristas que posee, y finalmente, cuenta para cada nodo, la cantidad de aristas que apuntan a él, de manera que si una página es muy apuntada por otras, su valor (importancia) aumenta considerablemente, dado que parece ser una página de interés para muchos otros. También es una función que toma en cuenta la importancia de los dueños de las aristas que apunta a él. A manera de ejemplo, podemos ver en la figura 1 de la página 2, podemos ver que el nodo B, es el más importante dado que recibe links de

[1]Nótese que se trata de un grafo dirigido, dado que las aristas tienen dirección, esto debido a que si, por ejemplo, en mi página web pongo un link a la página principal de google, eso no implica que en la página de principal de google, vaya a haber un link hacia mi página

muchos otros, mientras que el nodo C, es importante porque recibe el 100 % de los enlaces del nodo B. Finalmente los nodos sin nombre (morados), son nodos muy poco importantes dado que ningún otro nodo apunta a ellos.

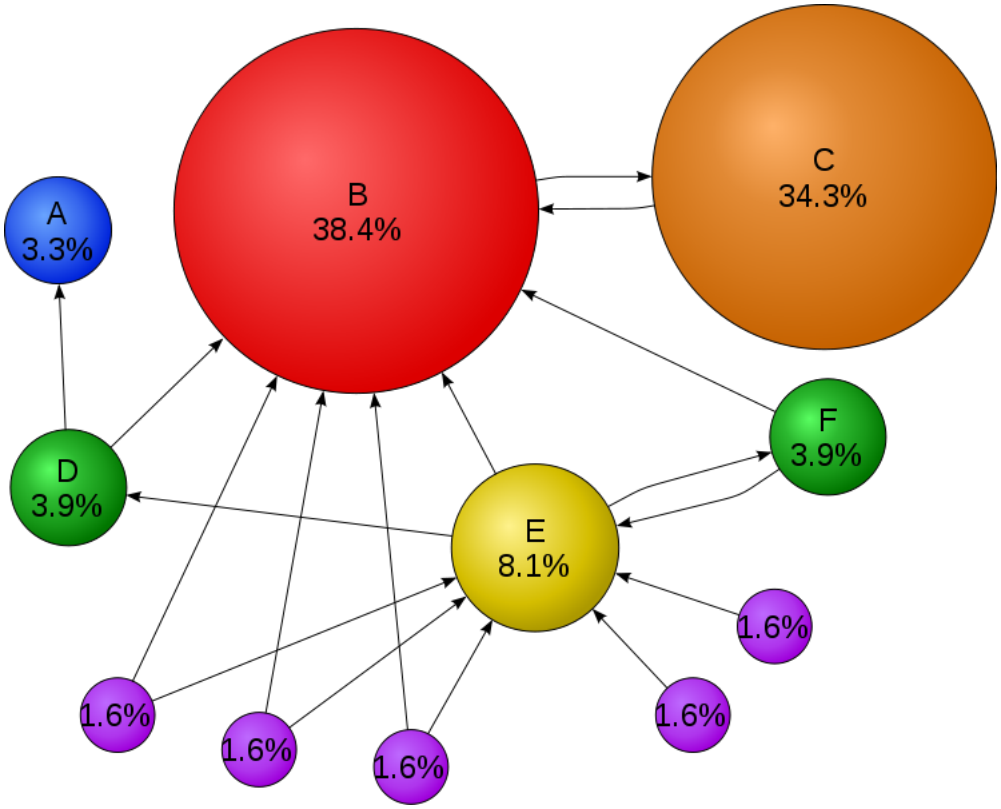


Figura 1: Page Rank

## 2.1. Algoritmo del PageRank

El algoritmo inicial del PageRank lo podemos encontrar en el documento original donde sus creadores presentaron el prototipo de Google: "The Anatomy of a Large-Scale Hypertextual Web Search Engine":

$\mathrm{PR}(A) = (1-d) + d\sum_{i=1}^{n} \frac{\mathrm{PR}(i)}{C(i)}$

Donde:

1. PR(A) es el PageRank de la página A.
2. d es un factor de amortiguación que tiene un valor entre 0 y 1.

3. PR(i) son los valores de PageRank que tienen cada una de las páginas i que enlazan a A.

4. C(i) es el número total de enlaces salientes de la página i (sean o no hacia A).

Algunos expertos aseguran que el valor de la variable d suele ser 0,85. Representa la probabilidad de que un navegante continúe pulsando links al navegar por Internet en vez de escribir una url directamente en la barra de direcciones o pulsar uno de sus marcadores y es un valor establecido por Google. Por lo tanto, la probabilidad de que el usuario deje de pulsar links y navegue directamente a otra web aleatoria es 1-d [2]

# 3. Descripción del problema

Su programa debe de poder programar de manera recursiva (tal y como la fórmula lo indica), el Page Rank para un grafo arbitrario.

## 3.1. Entradas y salidas del programa

Las entradas y salidas del programa deben de ser por medio de archivos. La entrada debe de ser un archivo que tiene una matriz de adyacencias, ejemplo:

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| C | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| E | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| F | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| H | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| I | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| J | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| K | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Y la salida sería:

1. A:0.03278149

[2]Wikipedia: http://es.wikipedia.org/wiki/PageRank

2. B:0.38440095
3. C:0.34291029
4. D:0.03908709
5. E:0.08088569
6. F:0.03908709
7. G:0.01616948
8. H:0.01616948
9. I:0.01616948
10. J:0.01616948
11. K:0.01616948

# 4. Características de la evaluación

## 4.1. Aspectos técnicos

1. Toda la programación debe de realizarse en Java o C++

## 4.2. Aspectos Administrativos

1. El proyecto puede realizarse en grupos de 3 personas

## 4.3. Evaluación

1. Montar el grafo correctamente (30 %)
2. Contar apropiadamente los enlaces entrantes y salientes (20 %)
3. Programar apropiadamente el PageRank

DISCLAIMER[3]

[3]DISCLAIMER: Todas las imágenes utilizadas en esta especificación fueron hechas por el usuario de wikipedia: 345Kai (http://en.wikipedia.org/wiki/User:345Kai)