

Proyecto Estructura de Datos y Algoritmos

Gestión de páginas web

Objetivos

Queremos crear una aplicación que gestione un número grande (miles, decenas de miles o cientos de miles) de páginas web. Los datos se han tomado de Web Data Commons - Hyperlink Graph 2012 - Download Instructions:

<http://webdatacommons.org/hyperlinkgraph/2012-08/download.html>

Una web está conectada con otras webs por medio de enlaces. Se debe usar el modelo de dominio siguiente:



Es decir, una web tiene enlaces salientes hacia otras web. De igual manera, una página web tiene asociadas una serie de palabras clave y cada palabra clave aparece en varias webs.

Los ficheros de páginas web tienen líneas de la forma:

Fichero "index-2024-25"	Fichero "pld-arc-1-N-2024-25"
0 ::: 0-00.pl	0 >>> 77783 ### 7854 ##### 795437 ### 862696
1 ::: 0-100.com.cn	1 >>>> 59887 ### 129783 ### 14334 ### 2136
2 ::: 0-100editions.net	2 >>> 413745 ##### 1141599
3 ::: 0-18.gr	3 >> 413745 ### 3452 ## 23456789 ### 333
4 ::: 0-200.com	4 >>>> 413745 ##### 356789 ## 76549
5 ::: 0-24-sex.de	...
6 ::: 0-24.ro	
7 ::: 0-3-6.com	
8 ::: 0-311.com	
9 ::: 0-360.com	
10 ::: 0-3ani.ro	
11 ::: 0-5-30.com	
12 ::: 0-5.co.il	
13 ::: 0-5ans.com	
14 ::: 0-60cartimes.com	
15 ::: 0-700.pl	
16 ::: 0-75.pl	
17 ::: 0-adult.net	
18 ::: 0-apr-creditcards.com	
19 ::: 0-art.co.uk	
20 ::: 0-brune-nue.com	
...	

El fichero de nombre “index” asocia un valor entero, empezando desde el 0, a cada página web. El fichero de nombre “pld-arc-1-N” tiene líneas de la forma “x >>> u ### v ## w”, indicando que en la web x hay enlaces (salientes) a las webs u, v, w.

El fichero “words.txt” contiene la lista de todas las posibles palabras clave:

```
...
credere
credibilities
credibility
credible
credibleness
credibly
credit
creditabilities
creditability
creditable
creditaleness
creditably
credited
crediting
...
```

Estas pueden ser algunas de las funcionalidades asociadas a las clases:

- private String **ident2String**(int x)
// Precondición: x es un valor entero ≥ 0
// Postcondición: devuelve la web asociada a x
Por ejemplo: `id2String(18)` → `0-apr-creditcards.com`
- ArrayList<String> **salientes**(String web)
// post: dado el nombre de una web, devuelve las páginas webs a las que hace referencia
- ArrayList<String> **websOrdenadas**()
// post: devuelve una lista ordenada alfabéticamente de las páginas web
// no modifica la lista de páginas web original
- ArrayList<String> **word2Webs**(String s)
// pre: “s” representa una palabra clave
// post: devuelve las webs que contienen la palabra “s”
Por ejemplo: `word2Webs(“money”)` →
 <007waystomakemoney.com, 1000moneymakingideas.com, ...>
- ArrayList<String> **web2Words**(String w)
// post: devuelve las palabras que aparecen en la web “w”
Por ejemplo: `web2Words(“1000moneymakingideas.com”)` →
 <money, making, ideas>
- ...

Actividad 1.

Diseño e implementación de un sistema que permite operaciones sobre la lista de páginas web y palabras clave.

Objetivo final: obtener un sistema que permitirá las siguientes operaciones de forma eficiente (se deben razonar los motivos de la eficiencia):

- Leer los datos desde los ficheros
- Buscar una página web
- Insertar una nueva página web
- Añadir un enlace saliente a una web
- Dada una web, devolver una lista con las páginas web accesibles desde ella
- Dada una palabra clave, devolver una lista las páginas web que contienen esa palabra
- Borrar una página web
- Guardar la lista de webs actualizada en ficheros
- Obtener una lista de páginas web ordenada alfabéticamente (esta operación no debe modificar la lista de páginas web, sino que debe devolver una nueva lista ordenada, de tipo ArrayList o LinkedList). Se debe implementar un algoritmo de ordenación, es decir, no se puede llamar a una función estándar de ordenación ya implementada.

Se deberá entregar:

- Programas que implementen lo pedido (ejecutados correctamente)
- Documentación describiendo el problema planteado, las alternativas examinadas, implementaciones, y eficiencia (ver ejemplos de documentación)

Fechas importantes:

Martes, 10-IX-2024 (primer subgrupo) y martes 17-IX-2024 (segundo subgrupo):

- Especificación inicial de la(s) clase(s) correspondientes al problema dado. Diagrama de clases principal
- Diseño de las pruebas
- Diseño de las estructuras de datos principales

Lunes, 7-X-2024, fecha límite entrega actividad 1. Se debe entregar un documento que contenga:

- Descripción general del problema
- Descripción general de las alternativas examinadas y las soluciones adoptadas, justificándolas en base a diferentes criterios, como por ejemplo eficiencia
- Diseño e implementación de los algoritmos
- Resultados empíricos (tiempo) de las pruebas realizadas

Además debéis rellenar y entregar el Checklist para verificar que habéis realizado todo lo que se os pide.