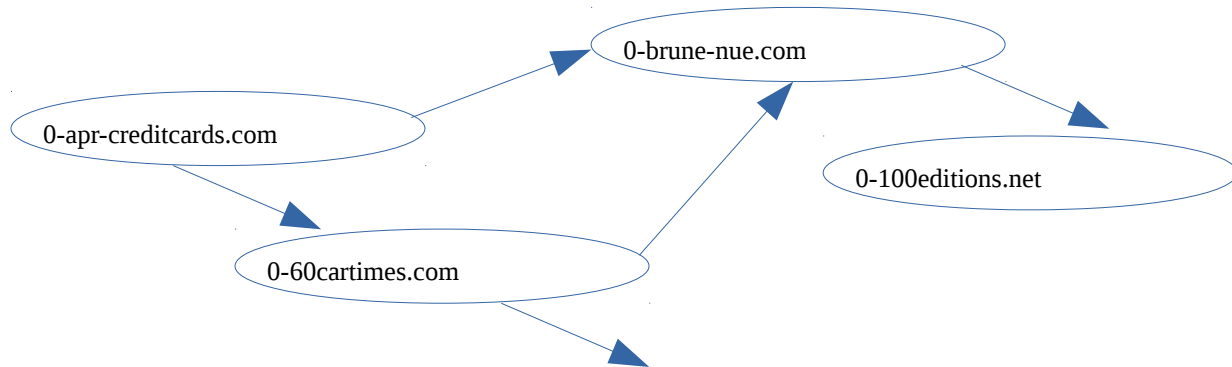


Actividad 3.

Objetivo final: obtener un sistema que permitirá las siguientes operaciones:

- boolean estanConectados(String a1, String a2)



El resultado será true si hay una cadena de relaciones que une a a1 y a2. Por ejemplo, la llamada a estanConectados("0-apr-creditcards.com", "0-100editions.net") dará true.

La estructura de datos que usaremos es similar al tipo Grafo de “Algorithms”, de Sedgewick y Wayne (página 552):

- HashMap<String, Integer> **th**

0-apr-creditcards.com	0
0-100editions.net	1
0-60cartimes.com	2
0-brune-nue.com	3
...	...

- String[] **keys**

0	0-apr-creditcards.com
1	0-100editions.net
2	0-60cartimes.com
3	0-brune-nue.com
...	...

- `ArrayList<Integer>[] adjList`

0	<2, 3>
1	<... >
2	<3>
3	<1>
4	<...>
5	<...>

```
public boolean estanConectados(String a1, String a2){
    Queue<Integer> porExaminar = new LinkedList<Integer>();

    int pos1 = th.get(a1);
    int pos2 = th.get(a2);
    boolean enc = false;
    boolean[] examinados = new boolean[th.size()];

    // COMPLETAR CÓDIGO

    return enc;
}
```

- OPCIONAL: `ArrayList<String> estanConectados(String a1, String a2)`

El resultado será una lista de relaciones desde a1 hasta a2, donde cada relación indica que esos 2 elementos están conectados. Por ejemplo, la llamada a `estanConectados("0-apr-creditcards.com", "0-100editions.net")` producirá:

<0-apr-creditcards.com>, <0-brune-nue.com>, <0-100editions.net>

Quiere decir que hay un enlace desde “0-apr-creditcards.com” a “0-brune-nue.com” y otro desde “0-brune-nue.com” a “0-100editions.net”.

Se deberá entregar:

- Programas que implementen lo pedido (ejecutados correctamente). **Se deberá demostrar que el programa funciona realmente con conjuntos de datos no triviales (es decir, procesando todo el fichero original).**
- Se deberá especificar de manera clara **el número de relaciones** (llamadas a `estanConectados`) que se calculan **en un periodo de tiempo** (un minuto, una hora, ...).
- Documentación describiendo el problema planteado, las alternativas examinadas, implementaciones, y eficiencia

26-XI-2017, fecha límite entrega actividad 3. 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 eficiencia
- Diseño e implementación de los algoritmos
- Este documento será una nueva versión (modificada y ampliada) de la documentación entregada en la actividad 1.

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