

Laboratorio 8: Tablas hash

Ejercicio 1.

Nos proporcionan un `HashMap` que guarda información de alumnos. Concretamente, la clave de la tabla es el código LDAP del alumno (de tipo `String`), y el valor es un `LinkedList` de enteros con los códigos de las asignaturas en las que está matriculado el alumno. Un ejemplo es el siguiente:

(clave: valor)

```
aamutxastegi001: [11167, 27539, 43700, 62101]
lzeberio003: [22654, 23132, 32649, 43700, 92177]
iuriarte012: [21350, 22654, 23132, 26031, 42692, 62101]
xandonegi002: [26031, 32020, 32643, 32649, 33627, 43700]
mgutierrez011: [22654, 33627, 43700, 62101, 83119, 93721]
ialbisu007: [27539]
egartzia043: [11167, 22654, 27539, 42692, 54543, 62101, 93721]
```

Se pide implementar el método *transformar*, que obtiene el `HashMap` inverso, es decir, en el que para cada asignatura se guarda la lista de alumnos matriculados en ella.

```
public HashMap<Integer, LinkedList<String>> transformar
    (HashMap<String, LinkedList<Integer>> alumnos){...}
```

Para el ejemplo, el resultado de *transformar* sería el siguiente:

(clave: valor)

```
32643: [xandonegi002]
42692: [iuriarte012, egartzia043]
21350: [iuriarte012]
32649: [lzeberio003, xandonegi002]
83119: [mgutierrez011]
26031: [iuriarte012, xandonegi002]
54543: [egartzia043]
92177: [lzeberio003]
27539: [aamutxastegi001, ialbisu007, egartzia043]
43700: [aamutxastegi001, lzeberio003, xandonegi002, mgutierrez011]
32020: [xandonegi002]
62101: [aamutxastegi001, iuriarte012, mgutierrez011, egartzia043]
```

```
93721: [mgutierrez011, egartzia043]
33627: [xandonegi002, mgutierrez011]
23132: [lzeberio003, iuriarte012]
22654: [lzeberio003, iuriarte012, mgutierrez011, egartzia043]
11167: [aamutxastegi001, egartzia043]
```

Ejercicio 2.

Llega la hora de realizar los exámenes globales y desde Decanato han pedido a los coordinadores de cada asignatura que realicen una solicitud indicando:

- Nombre de la asignatura
- Aulas donde les gustaría realizar el examen (la asignatura necesitaría todas las aulas indicadas)
- Hora de comienzo
- Hora de fin

Completa la implementación del método *verificarSolicitudes*, que dado un `ArrayList` de las solicitudes correspondientes al primer día de exámenes, devuelve un booleano que indica si es posible respetar todas las preferencias de aulas solicitadas. Evidentemente, en un mismo aula no puede haber solapamiento de dos asignaturas a la misma hora. **Supondremos que las solicitudes del `ArrayList` están ordenadas en base a la hora de fin del examen.**

Ejercicio 3.

Completa la implementación del método *emparejamientoCorrecto*. Este método recibe dos strings: un string de hashtags (cada hashtag comienza con el carácter #) y un string de palabras (no comienzan por #). Ambos strings tienen el mismo n° de elementos, y en cada string, los elementos están separados por espacios en blanco. Ejemplo:

Hashtags: #a #b #b #a

Palabras: cat dog dog cat

Consideraremos que al primer hashtag le corresponde a la primera palabra, al segundo la segunda, etc. El problema consiste en determinar si todos los elementos forman un emparejamiento correcto.

Un emparejamiento es correcto si a cada hashtag le corresponde una única palabra y a cada palabra un único hashtag. Dicho de otro modo, a un hashtag no le puede

corresponder más de una palabra y a una palabra no le puede corresponder más de un hashtag. En el ejemplo el emparejamiento es correcto, porque al hashtag *#a* le corresponde únicamente la palabra *cat*, y viceversa, y al hashtag *#b* le corresponde únicamente la palabra *dog*, y viceversa.

Si el emparejamiento es correcto, el método deberá mostrar por pantalla la palabra que le corresponde a cada hashtag. Si no es correcto, mostrará un mensaje indicando este hecho.

SALIDA ESPERADA: EJERCICIO 1

```
32643: [xandonegi002]
42692: [iuriarte012, egartzia043]
21350: [iuriarte012]
32649: [lzeberio003, xandonegi002]
83119: [mgutierrez011]
26031: [iuriarte012, xandonegi002]
54543: [egartzia043]
92177: [lzeberio003]
27539: [aamutxastegi001, ialbisu007, egartzia043]
43700: [aamutxastegi001, lzeberio003, xandonegi002, mgutierrez011]
32020: [xandonegi002]
62101: [aamutxastegi001, iuriarte012, mgutierrez011, egartzia043]
93721: [mgutierrez011, egartzia043]
33627: [xandonegi002, mgutierrez011]
23132: [lzeberio003, iuriarte012]
22654: [lzeberio003, iuriarte012, mgutierrez011, egartzia043]
11167: [aamutxastegi001, egartzia043]
```

SALIDA ESPERADA: EJERCICIO 2

Prueba 1:

EDA:

```
Aulas:[2.5, 2.6, 2.7]
De 9:00 a 11:00
```

PB:

```
Aulas:[1.9, 2.6]
De 12:00 a 14:00
```

MEI:

```
Aulas:[1.2, 2.6]
De 13:00 a 17:00
```

DBD:

```
Aulas:[1.9]
De 15:00 a 17:00
```

Es posible?: false

Prueba 2:

EDA:

```
Aulas:[2.5, 2.6, 2.7]
De 9:00 a 11:00
```

PB:

```
Aulas:[1.9, 2.6]
De 12:00 a 14:00
```

MEI:

```
Aulas:[1.2, 2.6]
De 14:01 a 17:00
```

DBD:

```
Aulas:[1.9]
De 15:00 a 17:00
```

Es posible?: true

SALIDA ESPERADA: EJERCICIO 3

Prueba 1:
Hashtags: #a #b #b #a
Palabras: cat dog dog cat
El emparejamiento es correcto.
#a->cat
#b->dog

Prueba 2:
Hashtags: #a #b #b #a
Palabras: cat cat dog cat
El emparejamiento no es correcto.

Prueba 3:
Hashtags: #a #b #a #a
Palabras: cat dog cow cat
El emparejamiento no es correcto.

Prueba 4:
Hashtags:
Palabras:
El emparejamiento es correcto.

Prueba 5:
Hashtags: #a #b #c #d #e #b #b #d #a
Palabras: ant bird cat dog elephant bird bird dog ant
El emparejamiento es correcto.
#c->cat
#d->dog
#e->elephant
#a->ant
#b->bird