

Tarea 4

Trabajo a entregar por GRUPO vía EDUCA. Indique claramente los integrantes de su grupo. El código debe poder probarse sin usar ningún IDE desde la línea de comandos. No incluya ninguna clase en package.

Ejercicios prácticos(40p)

Ejercicio 1 (10 puntos)

Escriba un programa que lea una cadena, y luego un diccionario de palabras (defina el diccionario en un archivo txt que debe ser adjuntado), y verifique si la cadena ingresada es una 'buena' contraseña. Aquí, se debe asumir que 'buena' significa:

- Tiene por lo menos 8 caracteres de longitud,
- No es una palabra del diccionario,
- No es una palabra del diccionario, seguido por un dígito del 0-9 (por ejemplo, 'hola5'),
- No son dos palabras del diccionario separadas por un dígito (por ejemplo, 'hola2mundo').

Se debe implementar de manera eficiente (es decir, no está permitida la fuerza bruta). Todas las operaciones de verificación se deben poder hacer en $O(1)$.

Obs: Se puede considerar que las operaciones length y split de Java se ejecutan en $O(1)$

Observación: Se debe observar la siguiente rúbrica:

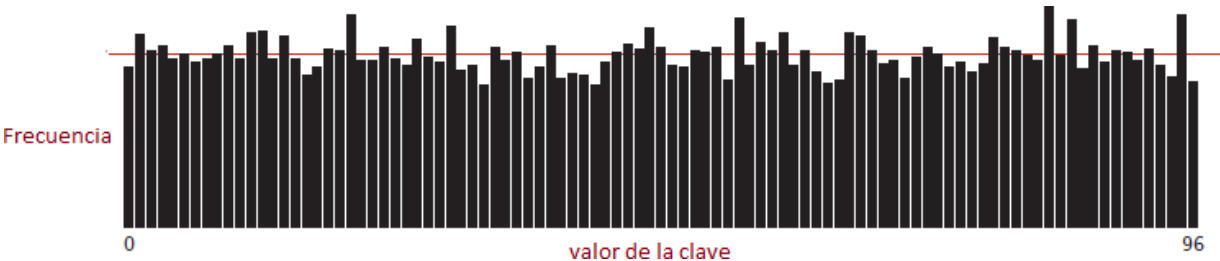
Rúbrica	Puntaje
Desarrollo correcto	4
Código fuente con comentarios	4
Provisión de datos de prueba	2

Ejercicio 2 (10p)

Verificar que la función de dispersión de cadenas que se muestra abajo, distribuye uniformemente las claves. Probar con los valores $R=31, 37, 32$ y 64 , además verificar con el esquema 2^k-1 (como el código mostrado) y aquel que utiliza todos los caracteres.

```
hash_value = 0;
for ( int k=1; k < s.length(); k*=2 )
    hash_value = hash_value * R + (s.charAt(k-1))
```

Realizar un gráfico de distribución de frecuencias para los diferentes valores de R . Comentar el resultado observado en dicho gráfico. Agrupe los valores en al menos 1000 franjas de forma que el gráfico pueda comprenderse bien. Abajo un gráfico ejemplo de frecuencias para 97 diferentes valores de clave.



Para las pruebas utilice la lista de palabras del diccionario español “*es_ES.dic*” (obtenido del sitio de OpenOffice (<http://www.openoffice.org>)). Algunas palabras tienen un indicador utilizado por el corrector ortográfico (<palabra><indicador>). Para el trabajo solo considerar <palabra> descartando todo lo sigue al carácter “/” incluyendo ese carácter. Por ejemplo, en el diccionario encontrará “abusar/RED”, solo considerar “abusar” para las pruebas. Este archivo está disponible en EDUCA.

Rúbrica	Puntaje
---------	---------

Desarrollo correcto	4
Gráfica de distribución de frecuencias	4
Provisión de datos de prueba	2

Ejercicio 3 (20p)

Implemente en Java las estructura de datos y los algoritmos de inserción, eliminación y búsqueda correspondiente a tablas de dispersión que utiliza dispersión abierta (usando listas enlazadas) y dispersión cerrada (usando exploración lineal, cuadrática y doble hashing). Para el caso de la dispersión cerrada tener en cuenta el *rehashing* para un factor de carga que se recibe como parámetro de la estructura de datos. Para el método de doble hashing utilice alguna descrita en el libro [CLRS2001] [CLRS2009]. Indique en el código exactamente cuál método de doble hashing utilizó, colocando la referencia del libro (ítem en el libro).

En el caso de la tabla de dispersión con exploración lineal utilizar para la eliminación, el método de eliminación perezosa.

Implemente usando *Generic* y utilice el método *hashCode()* de la clase *Object* para obtener el entero que represente al objeto. Cada clase de Java debe implementar *java.lang.Iterable*.

Pruebe la implementación insertando cadenas de la lista de palabras utilizadas en el ejercicio anterior. Para esta prueba indique cuál de las opciones de tablas de dispersión fue la mejor en cuanto a tiempo promedio de búsqueda, eliminación e inserción. Muestre sus resultados en una tabla (que debe ser parte del mismo código)

Finalmente evalúe la longitud de los agrupamientos (*clusters*) producidos en los métodos de dispersión cerrada. Para ello muestre por cada tipo de dispersión la distribución de frecuencia de los tamaños de agrupamientos que se generaron con el conjunto de datos indicado.

Rúbrica	Puntaje
Algoritmos de dispersión abierta	4
Algoritmos de dispersión cerrada	4
Utilización de método de eliminación perezosa	4
Utilización de Generics	4
Utilización de lista de palabras	2
Evaluación de longitud de agrupamientos	2