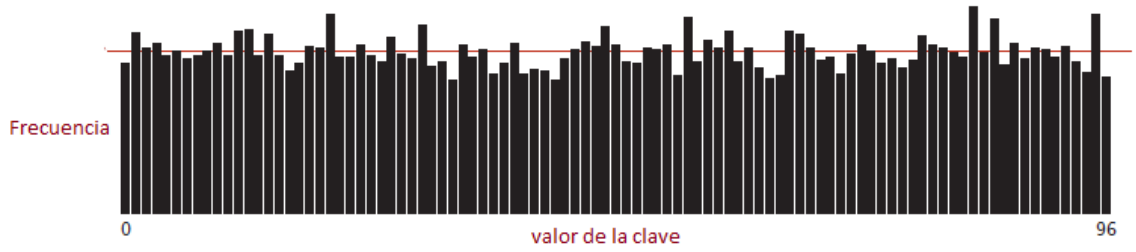
Algoritmos y ED III – 2018 – 1er Semestre - Tablas de dispersión.
Prof. Cristian Cappo – Tarea #5 – A entregar Viernes 23/marzo/2018 23:55 hs

Ejercicio 1 (10p)
Verificar que la función de dispersión de cadenas mostrada en la clase, 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 esta disponible en EDUCA.

Ejercicio 2 (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 cual 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 NO perezosa (mostrada en clase).

Implemente usando *Generic* además 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 1. Para esta prueba indique cual 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.

Ejercicio 3 (5p)

Pruebe empíricamente, haciendo experimentos con varios tamaños de N, que el número medio de celdas examinadas en una inserción con exploración lineal corresponde a la fórmula de la derecha. Teorema 19.2. Página 534 del libro de [WEISS2000].

$$\frac{1+\frac{1}{(1-\lambda)^2}}{2}$$

Puede utilizar el mismo conjunto de pruebas del ejercicio 2.

Ejercicio 4 (5p)

Por cada uno de los siguientes tipos de dato en Java (que estan incluidos), averigue y escriba en un documento como el API calcula su valor hash (es decir su representación de 32 bits). *Integer*, *Double*, *Float, String*, *java.util.Date*, *Long*, *java.math.BigInteger.* Sea lo más específico posible.