Laboratorio 1 - Hash

INS125 - Lenguajes de programación
Universidad Andrés Bello

7 de mayo de 2021

1. Recuperación de información

La recuperación de información (information retrieval) es una linea de investigación de las Ciencias de la computación que estudia los métodos existentes para recuperar información almacenada a través de ciertos criterior de entrada. Los buscadores como google son finalmente, un poderoso sistema de recuperación de información.

2. Tablas hash

Una tabla hash es una estructura de datos muy eficiente para realizar búsqueda de datos. Su funcionamiento consiste en asignar una clave calculada a cada elemento, llamada hash, obtenida a través de una función llamada función de hash.

Una función de hash evitará producir una misma clave para dos elementos distintos. Cuando esto ocurre, a este comportamiento se le conoce como colisión. Evitar colisiones requiere alto esfuerzo debido a que es necesario generar un segundo proceso en caso de estar en presencia de colisiones , el cual en general es computacionalmente costoso.

Las tablas hash tienen diversas aplicaciones, entre las cuales destacan su utilización en algoritmos de búsqueda para determinar rutas ya recorridas, sistemas de corrección ortográfica y sistemas de recuperación de información como buscadores o sistemas de clasificación.

3. Tarea

En esta tarea, a usted se le solicita implementar un sistema de recuperación de información el cual, dado una palabra, sea capaz de recuperar el contexto de aparición de esa palabra. Para ese propósito, usted deberá implementar un completo sistema de indexación y hashing el cual logre almacenar un *corpus* de texto con sus correspondientes índices y recuperar cual es el contexto de aparición de las palabras en consulta.

El funcionamiento del sistema debe ser el siguiente: Dado un archivo el cual contiene un corpus de texto, usted deberá generar un procedimiento el cual lo lea y **calcule el hash de cada palabra** (separados por espacios).

Luego usted utilizará el hash de cada una de palabras en un arreglo de largo 65.536 (2^{16}, cantidad máxima de palabras en cada corpus) y **utilizar cada hash calculado como un puntero a la última aparición de la palabra dentro del corpus**.

Para el cálculo del hash, considere una función que, por cada palabra w de largo l, calcule la función $hash(w)$ la cual debe estar definida por la siguiente formula, donde $w[0]$ corresponde al valor en ascii de la primera letra de la palabra w.

$$H(w) = w[0] * 10^0 + w[1] * 10^1 + w[2] * 10^2 + ... + w[l-1] * 10^{l-1}$$

$$hash(w) = \begin{cases} H(w) & \text{Si } H(w) < 65{,}536 \\ H(w)\,\%65536 & \text{Si } H(w) \geq 65{,}536 \end{cases}$$

Luego, en un arreglo A de tamaño 65.536 y de tipo `*char`, deberá almacenar en la posición $hash(w)$ un puntero hacia la posición del texto donde se encuentra la palabra correspondiente.

Si el hash calculado por la función $H(w)$ es mayor a 65.536, el hash para esa palabra $hash(w)$ corresponderá al resto de la división por 65.536 del hash $H(w)$ calculado.

Finalmente, la salida de su programa debe ser la palabra consultada, el hash de cada palabra y los siguientes 30 caracteres que están luego de la última aparición de la palabra.

4. Entrada y salida

Su programa, recibirá como entrada un archivo de texto que contiene R corpus de texto y una cantidad de palabras a consultar.

Como salida, su programa deberá generar un archivo de salida por cada uno de los corpus de texto, el cual contendrá los hash de las palabras consultadas y los 30 caracteres siguientes a la aparición de la palabra.

Si la palabra no existe en el corpus, su programa deberá simplemente agregar tres puntos suspensivos (...). También, si la última aparición de la palabra tiene menos de 30 caracteres restantes, debe agregar solo los caracteres restantes.

Considere también que por cada uno de los corpus procesados, debe liberar la memoria de su programa y este debe estar libre de memory leaks. En caso que su programa tenga memory leaks, con archivos grandes los cuales contengan muchos corpus de textos extensos,

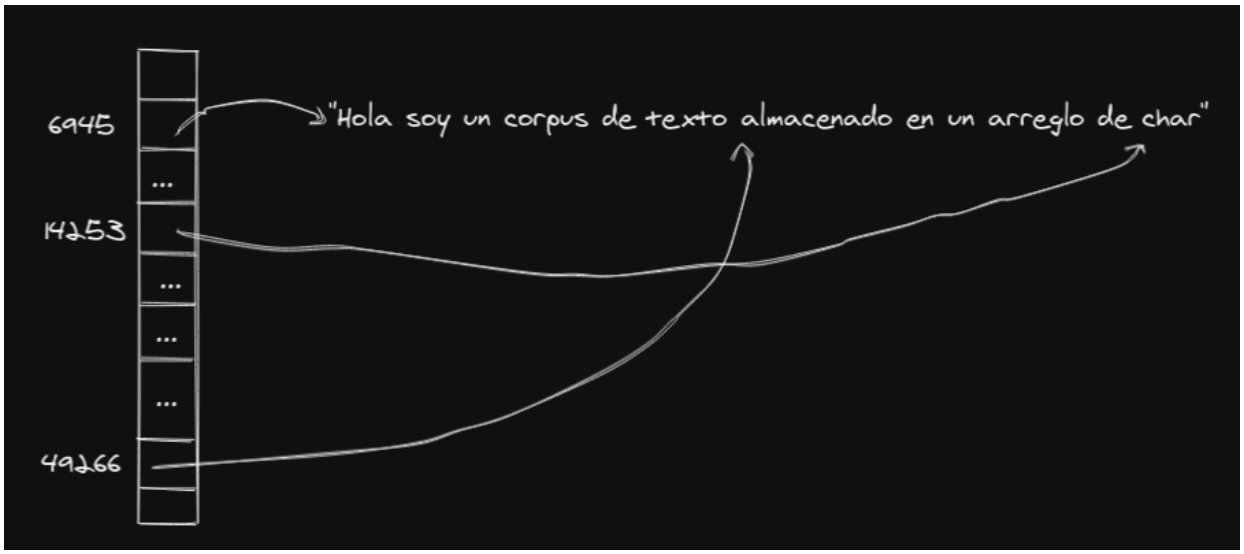

Figura 1: Ejemplo de funcionamiento del sistema. 6945 es el valor del hash para "Hola", 14253 es el valor de hash para çhar", etc.

su programa podría caer por problemas de memoria.

En caso que se produzcan colisiones, dado que dos palabras diferentes pueden generar el mismo hash, simplemente considere que son la misma palabra y su puntero apuntará a la última aparición de una de ellas. Su programa tampoco debe eliminar signos de puntuación de la cadena de entrada, es decir se consideran las palabras `'hola'`, `'hola.'` y `'hola!'` distintas, ya que estas generan distinto hash.

5. Ejemplo de archivo entrada.txt

```
The softmax function, also known as softargmax[1]:184 or normalized
exponential function,[2]:198 is a generalization of the logistic function
to multiple dimensions. It is used in multinomial logistic regression
and is often used as the last activation function of a neural network
to normalize the output of a network to a probability distribution
over predicted output classes, based on Luce's choice axiom.
3
softmax
normalized
multinomial
Instead of e, a different base b >0 can be used. If 0<b <1, smaller
input components will result in larger output probabilities, and decreasing
the value of b will create probability distributions that are more
concentrated around the positions of the smallest input values. Conversely,
if b >1, larger input components will result in larger output probabilities,
```

```
and increasing the value of b will create probability distributions
that are more concentrated around the positions of the largest input
values. Writing b = e [a] (for real v)[b] yields the expressions
3
smaller
probabilities
distributions
```

6. Ejemplo de archivos de salida

Cada linea del archivo de salida tendrá: la palabra consultada, el hash de la palabra consultada y los 30 caracteres restantes de la última aparición de la palabra; separados por tabulación.

Observe que cada palabra finaliza con tres puntos suspensivos (...). Si la palabra no está, solo deberá agregar los puntos suspensivos.

6.1. output1.txt

```
softmax 19270 softmax function, also known ...
normalized 46844 normalized exponential functi...
multinomial 44111 multinomial logistic regressi...
```

6.2. output2.txt

```
smaller 64324 smaller input components will...
probabilities 24110 ...
distributions 26340 distributions that are more c...
```

7. Instrucciones

- Fecha de entrega: Domingo 16 de mayo, 2021 a las 23:59.
- Método de entrega: Su repositorio privado creado a través del link de la tarea.
- Trabajo personal hecho en lenguaje C.
- Para comenzar su tarea, clone su repositorio y utilice el archivo `main.c` con código pre hecho. Ese código le servirá para separar las palabras existentes en el archivo de entrada. Puede editarlo con libertad.

- Su repositorio de la tarea solo debe contener un archivo el cual contendrá su programa, y este será llamado `main.c`. No incluya los archivos de entrada o salida en su repositorio. Utilice un archivo `.gitgnore` para no subirlo a su repositorio remoto.
- Su programa debe recibir y generar los archivos entrada.txt y salida.txt en la misma ubicación en la que se encuentra el programa ejecutable generado por la compilación.
- Solo puede utilizar bibliotecas estandar. Verifique que su programa compila al ejecutar el comando `gcc main.c -o main.o`.
- El proceso de revisión será automatizado. Es importante respetar el formato establecido para los archivos de entrada y salida. Este formato no es modificable. Por lo tanto, si su archivo de salida no corresponde al formato preestablecido, su calificación será deficiente.
- Las preguntas sobre la tarea deben ser formuladas como un Issue en el repositorio del laboratorio ubicado en el siguiente link `https://github.com/INS125/Laboratorio/issues`

8. Recomendaciones

- Se recomienda hacer commits parciales. Si su archivo de salida no contiene el contexto de aparición de la palabra (es decir, los siguientes 30 caracteres luego de su última aparición), puede obtener puntaje parcial si contiene la palabra y el hash.
- Si su programa no crea el archivo de salida o no compila, será evaluado con la nota mínima.
- En el repositorio oficial del laboratorio puede encontrar dos ejemplos de archivos de entrada y su correspondiente salida. `https://github.com/INS125/Laboratorio/`
- Es un trabajo personal. Es su responsabilidad cuidar su tarea.

9. Código de honor

Toda persona inscrita en este curso se compromete a:

- Actuar con honestidad, rectitud y buena fe frente a sus profesores y compañeros.
- No presentar trabajos o citas de otras personas como propias o sin su correspondiente citación, ya sea de algún compañero, libro o extraídos de internet como también a no reutilizar trabajos presentados en semestres anteriores como trabajos originales.
- No copiar a compañeros ni hacer uso de ayudas o comunicaciones fuera de lo permitido durante las evaluaciones.

Cualquier alumno o alumna que no respete el código de honor durante una evaluación (sea este la entrega de una tarea o el desarrollo de una prueba o control tanto durante la cátedra como el laboratorio) será evaluado con la nota mínima y será virtud de profesor, de acuerdo con la gravedad de la falta, las acciones siguientes a tomar.