

Prácticas de Algoritmia

Boletín 1. Hashing

Algoritmia

Juan Morales García

Grado en Ingeniería Informática



Índice de contenidos

1. Objetivo..... 1

2. Ejercicios..... 1

3. Entregables y puntuación..... 3

1. Objetivo

El objetivo de este boletín es que los alumnos analicen, diseñen, codifiquen y evalúen algoritmos de hashing dependiendo del tipo de solución utilizado para las colisiones.

2. Ejercicios

La empresa de compra-venta de coches “Sobre4Ruedas.es” desea optimizar la gestión de los datos de su negocio que actualmente están en el fichero autos.csv (adjunto a este boletín). Este fichero contiene 350 filas de datos sobre coches de segunda mano, donde cada fila representa:

- Primera columna: Matrícula del coche en el formato NNNNLLL, e.g. 2013CNR
- Segunda columna: Marca del coche
- Tercera columna: Modelo del coche
- Cuarta columna: Año del coche
- Quinta columna: Precio de tasación (en euros y sin decimales)

Las dos siguientes líneas muestran ejemplos de los datos incluidos en el fichero siguiendo el formato anterior:

9346QDV, Isuzu, Trooper, 1992, 9115

1333YYO, Mitsubishi, Fiero, 2005, 4715

La empresa quiere una aplicación que, dada la matrícula de un coche, devuelva toda la información de ese coche, y que la inserción y búsqueda de esos coches sean lo más eficiente posible. Para ello les han contratado y les piden aplicar la técnica Hash a este problema.

Todas las técnicas de hash deben soportar las operaciones de **insertar, buscar, eliminar y calcular factor de carga (% de ocupación de la tabla)**. Para cada inserción y búsqueda realizadas desde menú se debe mostrar por pantalla el número de intentos (accesos) a la tabla hash.

1. **(2,5 puntos)** Insertar los datos del fichero en una tabla hash de 350 elementos utilizando la técnica de **prueba lineal**.

Algoritmia

2. **(2,5 puntos)** Insertar los datos del fichero en una tabla hash de mínimo 350 elementos utilizando la técnica de **prueba dependiente de clave y asegurando un recorrido completo del algoritmo**. Se valorará la elección de una tabla que desperdicie el menor espacio posible pero mantenga una ocupación de la tabla por debajo del 50%.

3. **(3 puntos)** Elegir una de las dos siguientes alternativas:

a) Insertar los datos del fichero en una tabla hash inicial de 64 elementos utilizando la técnica de **prueba dependiente de la clave** que se redimensione cuando su factor de carga esté al 50%.

b) Insertar los datos del fichero en una tabla hash de 32 elementos utilizando la técnica de **encadenamiento**.

4. **(2 puntos)** Una vez codificados los ejercicios, se debe realizar una comparación empírica de los algoritmos teniendo en cuenta los siguientes ítems:

a) Tiempo empleado por el algoritmo en insertar todos los elementos (contad el tiempo desde que se inserta el primer elemento hasta que acaba el algoritmo de inserción).

b) Colisiones que se han producido en total en cada algoritmo (para el caso de la prueba dependiente de clave del ejercicio 3 no se debe tener en cuenta la redistribución al contar las colisiones).

Los resultados empíricos deben presentarse mediante una tabla o gráfica que indique claramente los valores obtenidos. Se deben justificar los resultados y comentar si los valores empíricos obtenidos se corresponden con lo estudiado en el tema de Hashing. Deben implementarse al menos dos algoritmos antes de realizar este ejercicio.

Además, se deben responder a las siguientes preguntas para los tres algoritmos

i. Explicar las decisiones y el código más relevante de cada ejercicio

ii. ¿Qué función $H(x)$ se ha utilizado? ¿Y qué función $G(x)$?

iii. ¿Cuántos accesos a la tabla hay que realizar para recuperar los registros con matrículas 4076XCH, 2180RCN y 6325PDS?

iv. Eliminar un registro que se haya insertado sin colisión. Probar a recuperar un registro que sí haya producido colisión con el que se acaba de eliminar. ¿Se recupera correctamente?

NOTAS:

- Los campos del fichero vienen separados por el carácter coma (.). Se adjunta código “leerFichero.c” como ayuda a la lectura del fichero de datos.
- Para el ejercicio 3 (encadenamiento) se puede utilizar el tipo “lista_dinámica” proporcionado por el profesor en la carpeta “Repaso C”.

3. Entregables y puntuación

- a. Memoria que contenga la información pedida en los ejercicios del boletín, junto a las decisiones más importantes tomadas para resolver cada ejercicio (NO INCLUIR EL CÓDIGO DE LOS PROGRAMAS. SIN MEMORIA NO SE CORREGIRÁN LAS PRÁCTICAS).
- b. Un proyecto de Dev-C++ / CodeBlocks que incluya los ficheros .c con el código de los programas y que esté listo para ser compilado y ejecutado (NO ENTREGAD LOS FICHEROS SIN PROYECTO O NO SE CORREGIRÁN LAS PRÁCTICAS).
- c. Fecha de entrega: **14 de Noviembre de 2021**
- d. Nota Máxima del boletín: 10 puntos
 - i. El 30% de la puntuación de cada ejercicio corresponderá a su documentación en la memoria que explique las decisiones más relevantes y al estilo de programación, teniendo en cuenta:
 - Comentarios adecuados en cantidad y calidad.
 - Refactorización del código, modularización en ficheros.
 - Tabulación correcta del código.
 - Uso de estructuras dinámicas en vez de estáticas, evitar el uso de variables globales, correcto paso de parámetros, etc.