



ESCUELA UNIVERSITARIA POLITÉCNICA
Departamento de Ciencias Politécnicas
Grado en Ingeniería Informática

Prácticas Algoritmia
Boletín 1. Hashing

Curso 2019/2020

Profesor: Andrés Muñoz

Boletín 1. Hashing

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 página web “FilmAmiguetes” desea optimizar la gestión de la base de datos de películas que los usuarios comentan en la web. Actualmente los datos de estas películas están en el fichero **films.csv** (adjunto a este boletín). Este fichero contiene 400 filas de datos sobre película, donde cada fila sigue el siguiente formato:

Año de estreno; Duración (en minutos); Título; Género; Popularidad

donde “Popularidad” es un valor entero entre 0 y 100 y los campos “Título” y “Género” son cadenas de texto. Las dos siguientes líneas muestran ejemplos de los datos incluidos en el fichero siguiendo el formato anterior:

1977;121;Star Wars;Science Fiction;44
1990;135;Cyrano de Bergerac;Drama;76

La empresa quiere una aplicación para insertar, buscar y eliminar películas de la manera 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 400 elementos utilizando la técnica de **prueba lineal**.
2. **(2,5 puntos)** Insertar los datos del fichero en una tabla hash de **mínimo 400** elementos utilizando la técnica de **prueba dependiente de clave y asegurando un recorrido completo del algoritmo**.
3. **(3 puntos)** Elegir una de las dos siguientes alternativas:
 - a) Insertar los datos del fichero en una tabla hash inicial de 32 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 en una tabla hash de 40 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 redispersión 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 campos se han usado como clave para la inserción? ¿Se han usado esos mismos campos para la búsqueda? Justifique las respuestas.**
- iii. **¿Qué función $H(x)$ se ha utilizado? ¿Y qué función $G(x)$?**
- iv. **¿Cuántos accesos a la tabla hay que realizar para recuperar los registros de las siguientes películas: “Octopussy”; “Cyrano de Bergerac”; “Hurricane”?**
- v. **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 punto y coma (;). Se adjunta código “leerFichero.c” como ayuda a la lectura del fichero de datos.
- La primera fila del fichero es la cabecera con los nombres de las variables, debe ser ignorada y no introducirla en la tabla hash.
- Se adjunta un código de ayuda para medir tiempos que el alumno puede utilizar en los ejercicios (va en la tarea).
- 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 DE CADA EJERCICIO POR SEPARADO O NO SE CORREGIRÁN LAS PRÁCTICAS).**
- c. **Fecha de entrega: 10 de Noviembre de 2019**
- d. **Nota Máxima del boletín: 10 puntos**
 - 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.