



ESCUELA UNIVERSITARIA POLITÉCNICA
Departamento de Ciencias Politécnicas
Grado en Ingeniería Informática

Prácticas Algoritmia
Boletín 1. Hashing

Curso 2018/2019

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 de casting de cine “Actores&Actrices.com” desea optimizar la gestión de los datos de su negocio que actualmente están en el fichero **listaactores.csv** (adjunto a este boletín). Este fichero contiene 500 filas de datos sobre actores, donde cada fila sigue el siguiente formato:

Apellidos; Nombre; Edad; Tiene Goya (S/N)*;Caché

donde los campos con * indica que sólo se toma una opción y el caché es la cantidad de dinero en € que cobra cada actor/actriz por hacer una película. Las dos siguientes líneas muestran ejemplos de los datos incluidos en el fichero siguiendo el formato anterior:

Bardem; Javier; 45; S; 20000

Morales; Gracita; 70; N; 60000

La empresa quiere una aplicación que, dado el nombre y el apellido de un actor o actriz, devuelva toda la información de dicha estrella de la gran pantalla, y que la inserción y búsqueda de estos elementos sea 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, recuperar, eliminar y calcular factor de carga (% de ocupación de la tabla)**. **Por cada inserción y búsqueda realizadas desde menú se debe mostrar por pantalla el número de intentos (accesos) a la tabla hash.**

1. **(2 puntos)** Insertar los datos del fichero en una tabla hash de 500 elementos utilizando la técnica de **prueba lineal**.
2. **(3 puntos)** Insertar los datos del fichero en una tabla hash de **mínimo 500** elementos utilizando la técnica de **prueba dependiente de clave y asegurando un recorrido completo del algoritmo**.
3. **Elegir una de las dos siguientes alternativas:**
 - a) **(2 puntos)** 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) **(2 puntos)** Insertar los datos del en una tabla hash de 50 elementos utilizando la técnica de **encadenamiento**.
4. **(3 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).

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 de los siguientes actores: “Javier Bardem”; “Dennis Gomez”; “Louise Jenkins”?
- 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?

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

NOTAS:

- Los campos del fichero vienen separados por el carácter coma (;).
- Se adjunta un código de ayuda “leerfichero.c” para leer el fichero “data.txt”. Se recomienda ejecutarlo tal cual se entrega para ver su funcionamiento y luego añadirlo al proyecto del alumno para realizar los ejercicios.
- Se adjunta un código de ayuda para medir tiempos que el alumno puede utilizar en los ejercicios
- Para el ejercicio 3 (encadenamiento) se puede utilizar el tipo “lista_dinámica” proporcionado por el profesor en la Unidad “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: 4 de Noviembre de 2018**
- d. **Nota Máxima del boletín: 10 puntos**
 - o 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.