



ESCUELA UNIVERSITARIA POLITÉCNICA
Departamento de Ciencias Politécnicas
Grado en Ingeniería Informática

Prácticas Algoritmia
Boletín 1. Hashing

Curso 2016/2017

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

Para todos los ejercicios se debe utilizar el fichero **data.txt** que se encuentra junto a este boletín. Este fichero contiene 100 filas de datos sobre personas, donde cada fila representa:

- Primera columna: DNI de la persona
- Segunda columna: Nombre de la persona
- Tercera columna: E-mail de la persona

Todas las técnicas de hash deben soportar las operaciones de **insertar, recuperar y eliminar y calcular factor de carga (% de ocupación de la tabla)**.

1. **(2 puntos)** Insertar los datos del fichero data.txt en una tabla hash de 100 elementos utilizando la técnica de **prueba lineal**.
2. **(3 puntos)** Insertar los datos del fichero data.txt en una tabla hash inicial de 32 elementos (habrá que ampliarla cuando se hayan insertado 32 registros) utilizando la técnica de **prueba dependiente de la clave**.
3. **(3 puntos)** Insertar los datos del fichero data.txt en una tabla hash de 10 elementos utilizando la técnica de **encadenamiento**.
4. **(2 puntos)** Una vez codificados los ejercicios, se debe realizar una comparación empírica de los tres 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 no se debe tener en cuenta la redispersión).

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. ¿Se insertan todos los elementos en la tabla hash?
- iv. ¿Cuántos accesos a la tabla hay que realizar para recuperar los registros con DNI 10637290, 21039483, y 89170831?
- 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?

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 (,).
- Para el ejercicio 3 (encadenamiento) se puede utilizar el tipo “lista_dinámica” proporcionado por el profesor en el carpeta “Repaso C”.

3. Entregables y puntuación

- 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).**
- 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**).**
- Fecha de entrega: 13 de Noviembre de 2016**
- 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.