



**UCAM**  
UNIVERSIDAD  
CATÓLICA DE MURCIA

# Boletín 1 - Hashing

Algoritmia

Andrés Muñoz Ortega

*Escuela Politécnica Superior*

*Convocatoria Febrero (2019/2020)*

**KEVIN ROBERTO CASTILLO MORALES**

**A05570618**

**MIGUEL ANGEL PELÁEZ DE LA FUENTE**

**32075606J**

domingo, 10 de noviembre de 2019

Universidad Católica San Antonio de Murcia - Tlf: (+34) 968 27 88 00 [info@ucam.edu](mailto:info@ucam.edu) - [www.ucam.edu](http://www.ucam.edu)

***COMPARACION EMPIRICA*** \_\_\_\_\_ **3**

## COMPARACION EMPIRICA

- 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).

Debido a la variación del tiempo en cada una de las ejecuciones se ha procedido a calcular el tiempo en un promedio de 5 ejecuciones por cada prueba, en la siguiente tabla y graficas se muestra los resultados obtenidos.

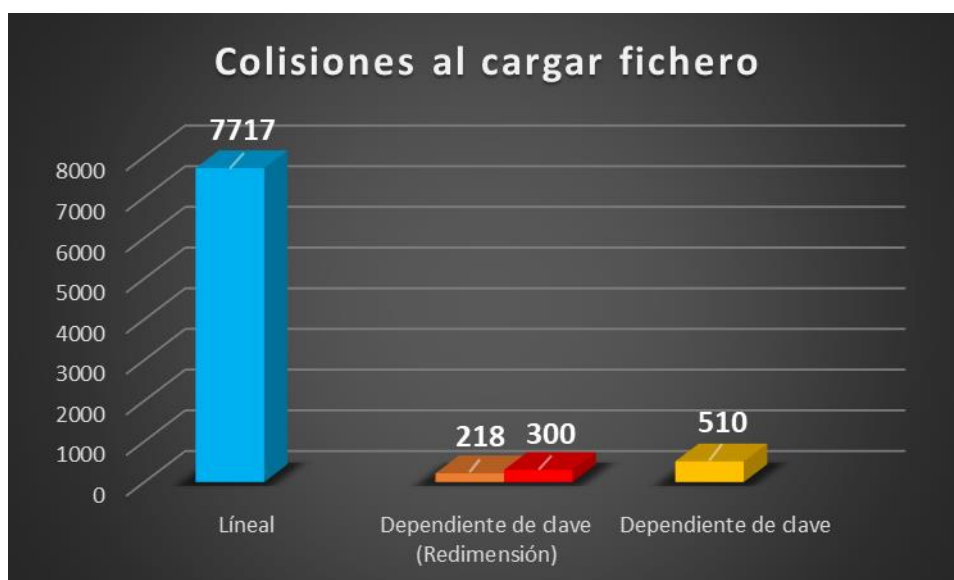
Prueba	Ejecución 1	Ejecución 2	Ejecución 3	Ejecución 4	Ejecución 5	Promedio
<b>Lineal</b>	0.000557	0.0005188	0.000611	0.0005082	0.0006408	0.00056716
<b>Dependiente de clave (Redimensión)</b>	0.0019479	0.0019079	0.0020405	0.001907	0.002575	0.00207566
<b>Dependiente de clave</b>	0.0007066	0.0003776	0.0004656	0.0003926	0.0007148	0.00053144



- 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).

La siguiente grafica muestra una comparación del numero de colisiones de cada algoritmo, en la prueba dependiente de la clave con redimensión se muestra el número de colisiones totales tomando en cuenta la redimensión y sin tomar en cuenta la redimensión.

Prueba	Colisiones	Contando la Redimensión
<i>Lineal</i>	7717	No Aplica
<i>Dependiente de clave (Redimensión)</i>	218	300
<i>Dependiente de clave</i>	510	No Aplica



Según los valores obtenidos de las graficas se puede ver con claridad en cuanto a tiempo de ejecución que la estructura dependiente de clave SIN redimensionar la tabla es mucho más efectiva, aunque la diferencia es muy poca en cuanto a la percepción humana, el ordenador si presencia un gran cambio en cuanto al tiempo que le toma cargar el archivo según la estructura utilizada. Sin embargo en cuanto al número de colisiones claramente la redimensión ayuda mucho ya que la tabla se hace cada vez más grande lo que disminuye el número de colisiones probables en el programa. Y tal como se ha de suponer la prueba lineal es la mas ineficiente de todas, sobre todo por el número de colisiones que presenta, aunque el tiempo de la redimensión es 4 veces el tiempo de la prueba lineal, en condiciones en que la redimensión se haga con un factor de carga más elevado la tabla presentaría una mejor eficiencia. En cuanto a tiempo de ejecución la prueba dependiente de clave sin redimensión se lleva el primer puesto, en cuanto a colisiones la prueba dependiente de clave con redimensión es la mejor opción.

**c) Explicar las decisiones y el código más relevante de cada ejercicio****PRUEBA LINEAL**

Una de las decisiones más relevantes fue saber si trabajar con memoria estática o con memoria dinámica, al final se optó por la dinámica para hacer mas eficiente en algunos aspectos el programa.

Se tiene una variable en el main para contar los accesos a la tabla hash y otra para contar las colisiones totales del programa, se pasan a las funciones como punteros para evitar el uso de variables globales.

Una confusión que se presentó al momento de utilizar la función G fue que para insertar tenía que devolver unos valores y para buscar otros diferentes en una misma condición, y fue muy complicado tomar la decisión de separar la función G en 2, una para insertar llamada "G" y la otra para buscar llamada "GBuscar".

Ambas funciones G implementan la recursividad al momento de encontrar una colisión para poder encontrar la posición deseada. La prueba lineal fue la que tuvo menos complicaciones al momento de tomar decisiones que influyeran de manera grande en el código.

**PRUEBA DEPENDIENTE DE CLAVE**

He introducido una variable que cuenta los accesos, que como su nombre indica va contando los accesos a la tabla hashing (buscar e insertar). También tenemos un contador de colisiones que nos servirá para la función "G" y para salir de la búsqueda sí no se encuentra. Finalmente usamos una variable "pos\_orig" (para saber la posición original) y que no se pierda ese dato y la variable "pos" que irá cambiando según las colisiones.

Para insertar va mirando el estado de "key" hasta que encuentra una posición libre para insertar los datos. Para buscar va comparando el "id" (que se encuentra en el "key") de cada posición hasta que lo encuentra o hasta que encuentre una posición limite o supere las 511 colisiones.

**PRUEBA DEPENDIENTE DE CLAVE (REDIMENSION)**

La mayor parte de la implementación hash es la misma que la de la prueba lineal.

Para hacer la redimensión se utiliza una tabla dinámica, ya que facilitaría mucho la redimensión, la función "redimensionar" utilizada se auxilia de una tabla extra con el nuevo tamaño a utilizar, en este auxiliar se almacenan los valores ya guardados en una nueva posición y la función devuelve un puntero a la nueva tabla hash, y limpia la memoria de la tabla anterior ya que no va a ser utilizada de nuevo.

Se creo una función D que genere un numero dependiendo de la clave, se utilizó una función extra ya que era más fácil llamar a la función en las partes que se necesita, que usar código duplicado.

El número de accesos a la tabla y el total de colisiones se pasa como puntero para evitar el uso de variables globales y así poder aumentar los contadores cada vez que deben ser aumentados.

También fue relevante elegir los puntos en donde llamar a la función redimensionar.

d) ¿Qué campos se han usado como clave para la inserción? ¿Se han usado esos mismos campos para la búsqueda? Justifique las respuestas.

Se ha utilizado como clave el año de estreno, la duración en minutos y la popularidad de la película, concatenados en ese mismo orden, hemos optado por estas opciones ya que es muy difícil que 2 películas coincidan en los 3 campos concatenados. Se usan estos 3 valores tanto para insertar como para buscar una película, ya que no tendría sentido alguno buscar una película por una clave diferente a la usada para la inserción.

Se guardan los valores como String en variables auxiliares y separados para poder concatenarlos, al tener el String concatenado se convierte a entero para poder almacenarlo como key y poder calcular las posiciones de cada película.

e) ¿Qué función  $H(x)$  se ha utilizado?

En los 3 casos se utiliza la misma función Hash, la típica utilizada en los ejercicios de clase, optamos por elegir esta y no una diferente para poner a prueba la eficiencia de esta función en diferentes tipos de manejo de colisiones.

$H(K) = K \% N$ . (N tamaño de la tabla hash), N varia en el ejercicio de redimensión de la tabla.

Nos hemos percatado que tanto en la prueba lineal como en la prueba dependiente de clave con esta función hash, los valores que tienen una similitud en los últimos dígitos de su key, colisionan entre sí.

f) ¿Y qué función  $G(x)$ ?

En la prueba lineal se utiliza la función  $G(H(K)) = (H(K) + \text{NumColisiones}) \% N$ . (N tamaño de la tabla hash).

En la función G uso los siguientes parámetros: el key, la posición original, el tamaño total y el contador de colisiones. Dentro de la función G usamos "d" (como en los apuntes) que se calcula dividiendo el "key" por el tamaño. Posteriormente si "d" es par se suma uno para que no se haga bucle y finalmente se aplica la fórmula que suma la posición con la multiplicación de "d" y el número de colisiones. En el "return" devuelve el módulo de la posición calculada anteriormente con el tamaño.

En la prueba dependiente de la clave con redimensión se utiliza la función  $G(H(K)) = (H(K) + (D \times \text{NumColisiones})) \% N$ . Donde D es un valor dependiente de la clave con la siguiente formula  $D = K / N$  (N tamaño de la tabla hash). Y si  $D \% 2 = 0$  se le incrementa 1 a su valor, ya que la tabla hash siempre tendrá un tamaño potencia de 2 en cada redimensión, y para asegurar el recorrido completo de la tabla D tiene que ser un valor impar.

- g) ¿Cuántos accesos a la tabla hay que realizar para recuperar los registros de las siguientes películas: “Octopussy”; “Cyrano de Bergerac”; “Hurricane”?

**Se proporciona los datos para buscar las películas solicitadas:**

**Octopussy:** *año 1983, duración 140, popularidad 68.*

**Cyrano de Bergerac:** *año 1990, duración 135, popularidad 76.*

**Hurricane:** *año 1979, duración 115, popularidad 8.*

### **PRUEBA LINEAL**

**Octopussy:** *1 acceso a la tabla.*

**Cyrano de Bergerac:** *3 accesos a la tabla.*

**Hurricane:** *169 accesos a la tabla.*

### **PRUEBA DEPENDIENTE DE CLAVE**

**Octopussy:** *1 acceso a la tabla.*

**Cyrano de Bergerac:** *3 accesos a la tabla.*

**Hurricane:** *3 accesos a la tabla.*

### **PRUEBA DEPENDIENTE DE CLAVE (REDIMENSION)**

**Octopussy:** *1 acceso a la tabla.*

**Cyrano de Bergerac:** *1 acceso a la tabla.*

**Hurricane:** *1 acceso a la tabla.*

- h) 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?

### PRUEBA LINEAL

Películas que colisionan en la prueba lineal

```
Titulo: Cool World
Año de estreno: 1992
Duracion: 101
Genero: Drama
Calificacion: 44
```

```
Titulo: Star Wars
Año de estreno: 1977
Duracion: 121
Genero: Science Fiction
Calificacion: 44
```

Si eliminas Cool World (No colisiona), puedes encontrar StarWars sin ningún problema. Y lo mismo con el resto de películas que colisionan entre si.

### PRUEBA DEPENDIENTE DE CLAVE

Películas que colisionan en la prueba dependiente de clave.

```
Titulo: Tie Me Up! Tie Me Down!
Año de estreno: 1990
Duracion: 111
Genero: Comedy
Calificacion: 68
```

```
Titulo: Raiders of the Lost Ark
Año de estreno: 1981
Duracion: 116
Genero: Action
Calificacion: 8
```

Si eliminas Tie Me Up! Tie Me Down! (No colisiona), puedes encontrar Raiders of the Lost Ark sin ningún problema.

### PRUEBA DEPENDIENTE DE CLAVE (REDIMENSION)

Películas que colisionan en la prueba dependiente de clave con redimensión.

```
Titulo: Fast Times at Ridgemont High
Año de estreno: 1982
Duracion: 92
Genero: Comedy
Calificacion: 65
```

```
Titulo: Empire Strikes Back, The
Año de estreno: 1980
Duracion: 124
Genero: Science Fiction
Calificacion: 33
```

Si eliminas Fast Times at Ridgemont High (No colisiona), puedes encontrar Empire Strikes Back sin ningún problema.

No hay problemas al buscar un elemento que se haya insertado con colisión porque en la búsqueda si encuentra un elemento que ha sido eliminado continúa buscando en la siguiente posición hasta encontrar el elemento buscado, o hasta recorrer todas las posiciones posibles sin encontrar ningún resultado.