



# HASH

03/07/2020



# Concepto General



## Concepto General

- Son estructuras que permiten almacenar elementos en una posición.



## Concepto General

- Son estructuras que permiten almacenar elementos en una posición.
- Dicha posición está determinada por su clave.



## Concepto General

- Son estructuras que permiten almacenar elementos en una posición.
- Dicha posición está determinada por su clave.
- Suelen implementarse en vectores ya que el acceso a las posiciones debe ser directo.



## Concepto General

- Son estructuras que permiten almacenar elementos en una posición.
- Dicha posición está determinada por su clave.
- Suelen implementarse en vectores ya que el acceso a las posiciones debe ser directo.
- Insertar, eliminar y buscar tienen complejidad constante.



**¿Cómo determino la clave de un elemento?**



## ¿Cómo determino la clave de un elemento?

- Obtenemos la clave, a partir de uno o más campos del elemento.





## ¿Cómo determino la clave de un elemento?

- Obtenemos la clave, a partir de uno o más campos del elemento.
- Las claves de los elementos no siempre son numéricas, ni correlativas, etc.



## ¿Cómo determino la clave de un elemento?

- Obtenemos la clave, a partir de uno o más campos del elemento.
- Las claves de los elementos no siempre son numéricas, ni correlativas, etc.
- En esos casos debemos transformarlas a posiciones de un vector.



## ¿Cómo determino la clave de un elemento?

- Obtenemos la clave, a partir de uno o más campos del elemento.
- Las claves de los elementos no siempre son numéricas, ni correlativas, etc.
- En esos casos debemos transformarlas a posiciones de un vector.
- Esta transformación se realiza a través de una función de hasheo.



# Función de Hasheo



## Función de Hasheo

- Nos permite, a partir de un elemento, determinar la posición que ocupará en nuestro hash.



## Función de Hasheo

- Nos permite, a partir de un elemento, determinar la posición que ocupará en nuestro hash.
- Debe ser la misma utilizada para todos los elementos.



## Función de Hasheo

- Nos permite, a partir de un elemento, determinar la posición que ocupará en nuestro hash.
- Debe ser la misma utilizada para todos los elementos.
- Debe dar valores lo más aleatorio posible, para las posiciones disponibles del hash.



## Función de Hasheo

- Nos permite, a partir de un elemento, determinar la posición que ocupará en nuestro hash.
- Debe ser la misma utilizada para todos los elementos.
- Debe dar valores lo más aleatorio posible, para las posiciones disponibles del hash.
- El cálculo no debe ser complejo.





# Función de Hasheo

¿Cuántas funciones de hasheo existen?



## Función de Hasheo - Restas sucesivas

Cuando se tienen huecos de tamaño conocido entre los elementos, puede aprovecharse ese espacio en nuestro hash, realizando los corrimientos determinados por el problema.

Supongamos los cursos 75.40 y 75.41 (siendo 75 el departamento y 4X el código de la materia), en los cuales no hay más de 200 alumnos.



## Función de Hasheo - Restas sucesivas

Las claves para cada alumno podrían estar dadas por  
departamento+materia+id\_alumno:

7540001

7540002

...

7540199

7541001

7541002

...



## Función de Hasheo - Restas sucesivas

Se desperdiciaría mucho espacio (800 posiciones entre materia + el corrimiento inicial) si no se utiliza una función de hasheo acorde.

7540001 ->  $7540001 - 7540000 = 1$  -> Posición en el hash

7540002

...

7540199

7541001

7541002

...



## Función de Hasheo - Restas sucesivas

Se desperdiciaría mucho espacio (800 posiciones entre materia + el corrimiento inicial) si no se utiliza una función de hasheo acorde.

7540001 ->  $7540001 - 7540000 = 1$  -> Posición en el hash

7540002 ->  $7540002 - 7540000 = 2$

...

7540199 ->  $7540199 - 7540000 = 199$

7541001

7541002

...



## Función de Hasheo - Restas sucesivas

Se desperdiciaría mucho espacio (800 posiciones entre materia + el corrimiento inicial) si no se utiliza una función de hasheo acorde.

7540001 ->  $7540001 - 7540000 = 1$  -> Posición en el hash

7540002 ->  $7540002 - 7540000 = 2$

...

7540199 ->  $7540199 - 7540000 = 199$

7541001 ->

7541002

...



## Función de Hasheo - Restas sucesivas

Se desperdiciaría mucho espacio (800 posiciones entre materia + el corrimiento inicial) si no se utiliza una función de hasheo acorde.

7540001 ->  $7540001 - 7540000 = 1$  -> Posición en el hash

7540002 ->  $7540002 - 7540000 = 2$

...

7540199 ->  $7540199 - 7540000 = 199$

7541001 ->  $7541001 - 7541000 = 1$

7541002

...



## Función de Hasheo - Restas sucesivas

Se desperdiciaría mucho espacio (800 posiciones entre materia + el corrimiento inicial) si no se utiliza una función de hasheo acorde.

7540001 ->  $7540001 - 7540000 = 1$  -> Posición en el hash

7540002 ->  $7540002 - 7540000 = 2$

...

7540199 ->  $7540199 - 7540000 = 199$

7541001 ->  $7541001 - 7541000 = 1$  -> CHAN

7541002

...





## Función de Hasheo - Restas sucesivas

Se desperdiciaría mucho espacio (800 posiciones entre materia + el corrimiento inicial) si no se utiliza una función de hasheo acorde.

7540001 ->  $7540001 - 7540000 = 1$  -> Posición en el hash

7540002 ->  $7540002 - 7540000 = 2$

...

7540199 ->  $7540199 - 7540000 = 199$

7541001 ->  $7541001 - 7541000 + 200 = 201$

7541002

...



## Función de Hasheo - Restas sucesivas

Se desperdiciaría mucho espacio (800 posiciones entre materia + el corrimiento inicial) si no se utiliza una función de hasheo acorde.

7540001 ->  $7540001 - 7540000 = 1$  -> Posición en el hash

7540002 ->  $7540002 - 7540000 = 2$

...

7540199 ->  $7540199 - 7540000 = 199$

7541001 ->  $7541001 - 7541000 + 200 = 201$

7541002 ->  $7541002 - 7541000 + 200 = 202$

...



## Función de Hasheo - Aritmética Modular

Es quizás la más intuitiva, se usa cuando sabemos que no hay huecos en nuestro dominio.

Consiste en, dada una clave, calcular el resto de dividirlo por la cantidad de posiciones disponibles.

Se suele utilizar cantidad de posiciones **primas** para favorecer la dispersión de la función.



## Función de Hasheo - Aritmética Modular

Durante el parcialito tuvieron que elegir que sobre les tocaba, supongamos que hay 3 alumnos, y 5 correctores. ¿Que dato del alumno podrían utilizar para determinar qué corrector les toca?

- 89364
- 89546
- 89652



## Función de Hasheo - Aritmética Modular

Durante el parcialito tuvieron que elegir que sobre les tocaba, supongamos que hay 3 alumnos, y 5 correctores. ¿Que dato del alumno podrían utilizar para determinar qué corrector les toca?

- $89364 \rightarrow 89364 \% 5 = 4$
- $89546 \rightarrow 89546 \% 5 = 1$
- $89652 \rightarrow 89652 \% 5 = 2$



## Función de Hasheo - Mitad del Cuadrado

Consiste en elevar al cuadrado la clave y tomar como lugar en el hash, los dígitos que ocupan una determinada posición, siempre la misma. El número de dígitos a tomar está determinado por el tamaño del hash.

Si tenemos un hash de 1000 posiciones y elegiremos las posiciones 3, 5 y 7 del cuadrado de la clave:

89364 ->  $89364^2$  -> 7985924496 -> Posición 498



## Función de Hasheo - Truncamiento

Consiste en ignorar parte de la clave y utilizar la parte restante directamente como índice.

Supongamos que tenemos claves de 5 dígitos, y 100 posiciones en nuestro hash.

Tomaremos los dígitos 1 y 3

89364 -> Posición 69



## Función de Hasheo - Plegamiento

Consiste en la división de la clave en diferentes partes y su combinación de un modo conveniente para obtener el índice.

Se desprecian los dígitos más significativos que se obtengan del arrastre o acarreo.

Supongamos un hash de 1000 posiciones y claves de 6 dígitos.

889364 -> 889 + 364 -> 1253 -> Posición 253





## Colisiones

Cuando la función de hash devuelve, para un elemento, una posición que ya está ocupada, se dice que hay una colisión.

Lo cierto es que no podemos asegurar que el hash va a tener lugar para TODOS los elementos que existan...

Sería como abrir un restaurant con capacidad para los 7,53 miles de millones de personas del mundo, eso no va a pasar.



# Colisiones

Al crear estructuras se hacen estimaciones y se espera que esas estimaciones se ajusten lo mejor posible a la realidad y los sistemas respondan de la mejor manera.

Pero puede suceder que no...

¿Cómo podríamos resolver las colisiones? ¿Qué harían si tienen un estacionamiento y la posición donde deben dejar el auto está ocupada?



## Hash Cerrado

Se dice que un hash es cerrado, cuando resuelve las colisiones ubicando a los nuevos elementos dentro del mismo hash.

Para resolver las colisiones se suelen utilizar 2 métodos:

- Zona de desborde.
- Exploración de posiciones.



## Hash Cerrado - Zona de Desborde

Si se determina que nuestra función de hash devuelve valores entre 0 y  $N-1$ , entonces nuestro hash deberá almacenar  $M$  posiciones.

Siendo  $M$  mayor que  $N$  y las posiciones entre  $N$  y  $M$  corresponden a la zona de desborde.



## Hash Cerrado - Exploración de Posiciones

Dado una colisión se dispone que la posición a ocupar será por:

- Prueba lineal: Próxima disponible,  $p+1$ ,  $p+2$ ...
- Prueba cuadrática: Se inspeccionarán las posiciones  $p+1^2$ ,  $p+2^2$ ,  $p+3^2$ ,  $p+4^2$ ...



## Hash Abierto

Se dice que un hash es abierto, cuando resuelve las colisiones ubicando a los nuevos elementos dentro de una lista de elementos por cada posición del hash.

Para resolver las colisiones, simplemente los elementos se insertan en la lista que le corresponde a su clave luego de aplicar la función de hasheo.