



Universidad Tecnológica Metropolitana

Escuela De informática

Busqueda Mediante Hashing

Integrantes:

David Martínez Rivas
Felipe Canales Saavedra
Javier Reyes Gonzalez
Fernando Rubilar Zepeda

Profesor:

Alejandro Reyes

Chile - 27 de abril de 2015

Índice general

.1.	Introducción	2
.2.	Historia	3
.3.	Función hash	4
	.3.1. Ventajas	6
	.3.2. Desventajas	6
.4.	Aplicación	7
.5.	Conclusión	8

.1. Introducción

En informática, hash se refiere a una función o método para generar claves o llaves que representan de manera casi unívoca a un documento, registro, archivo, etc., resumir o identificar un dato a través de la probabilidad, utilizando una función hash o algoritmo hash. Un hash es el resultado de dicha función o algoritmo

.2. Historia

Orígenes del término El término hash proviene, aparentemente, de la analogía con el significado estándar (en inglés) de dicha palabra en el mundo real: picar y mezclar. Donald Knuth cree que H. P. Luhn, empleado de IBM, fue el primero en utilizar el concepto en un memorándum fechado en enero de 1953. Su utilización masiva no fue hasta después de 10 años. En el algoritmo SHA-1, por ejemplo, el conjunto de partida de la función es dividido en palabras que son mezcladas entre sí utilizando funciones matemáticas seleccionadas especialmente. Se hace que el rango de valores que puede devolver la función sea de longitud fija: 160 bits utilizando la adición modular.

.3. Función hash

Es una función para resumir o identificar probabilísticamente un gran conjunto de información, dando como resultado un conjunto imagen finito generalmente menor. Varían en los conjuntos de partida y de llegada y en cómo afectan a la salida similitudes o patrones de la entrada.

Para encontrar la *función hash* no existe una regla que permita determinar cuál será la función más apropiada para generar un conjunto de claves que aseguren la máxima uniformidad en la distribución de las mismas. Algunas de las funciones hash más utilizadas son las siguientes:

- Función cuadrada.
- Función plegamiento.
- Función truncamiento.

La función módulo o por división toma el residuo de la división entre la clave y el total de elementos de la estructura, generando la siguiente fórmula:

$$\text{dirección} = (\text{clave} \% \text{total elementos})$$

Para lograr una mayor uniformidad en la distribución de los elementos, se debe buscar que el valor que se usa en el total de elementos sea un número primo más cercano al tamaño de la estructura.

Ejemplo. Si tenemos un total de 100 elementos y dos claves que sean 7259 y 9359, las direcciones generadas son las siguientes:

$$\text{dirección} = (7259 \% 100) = 59$$

$$\text{dirección} = (9359 \% 100) = 59$$

Estos dos casos generan una colisión, ya que los dos números no se pueden asignar dentro de la misma dirección en la estructura, para evitar la colisión, se cambia el valor de 100 por el número primo más cercano a él, en este caso sería un 97, lo que generaría las siguientes direcciones:

$$\text{dirección} = (7259 \% 97) = 81$$

$$\text{dirección} = (9359 \% 97) = 47$$

La función cuadrada como su nombre lo indica eleva al cuadrado la clave y del resultado, se toman los dígitos centrales como la dirección. El número de dígitos a tomar se determina del por el rango del índice de toda la estructura. La fórmula hash es la siguiente: dirección = dígitos centrales (clave²)
Ejemplo. Si tenemos un total de 100 elementos y dos claves que sean 7259 y 9359, las direcciones generadas son las siguientes:

$$\text{dirección} = \text{dígitos centrales } (7259^2) = 52693081 = 93$$

$$\text{dirección} = \text{dígitos centrales } (9359^2) = 87590881 = 90$$

Como el rango de claves es de 1 a 100 se toman dos dígitos centrales.

La función plegamiento divide la clave en partes de igual número de dígitos (la última puede tener menos dígitos), tomando como dirección los dígitos menos significativos, después de realizar una operación entre las partes, ya sea una serie de sumas o de multiplicaciones. La fórmula sería la siguiente:

$$\text{dirección} = \text{dígitos menos significativos (suma de partes)}$$

$$\text{dirección} = \text{dígitos menos significativos (multiplicación de partes)}$$

Ejemplo. Si tenemos un total de 100 elementos y dos claves que sean 7259 y 9359, las direcciones generadas son las siguientes:

$$\text{dirección} = \text{dígitos menos significativos } (72 + 59) = \text{dígitos menos significativos } (131) = 31$$

$$\text{dirección} = \text{dígitos menos significativos } (93 + 59) = \text{dígitos menos significativos } (152) = 52$$

Como el rango de claves es de 1 a 100 se toman dos dígitos para las particiones y para la dirección.

La función truncamiento toma algunos de los dígitos de las claves y forma con ellos una dirección. La elección de los dígitos es arbitraria, podrían tomarse los de las posiciones pares o impares para con ellos generar la dirección donde se almacenara la clave, uniendo los dígitos de izquierda a derecha o de derecha a izquierda, su fórmula es la siguiente:

$$\text{dirección} = \text{elegir dígitos (unión dígitos)}$$

Ejemplo. Si tenemos un total de 100 elementos y dos claves que sean 7259 y 9359, las direcciones generadas son las siguientes:

$$\text{dirección} = \text{elegir dígitos } (7, 5) = 75$$

$$\text{dirección} = \text{elegir dígitos } (9, 5) = 95$$

Para este caso se tomaron los dígitos impares y se unieron de izquierda a derecha.

.3.1. Ventajas

Se pueden usar los valores naturales de la llave, puesto que se traducen internamente a direcciones fáciles de localizar Se logra independencia lógica y física, debido a que los valores de las llaves son independientes del espacio de direcciones No se requiere almacenamiento adicional para los índices.

.3.2. Desventajas

El archivo no esta clasificado No permite llaves repetidas Solo permite acceso por una sola llave Costos Tiempo de procesamiento requerido para la aplicación de la función hash

.4. Aplicación

- Es posible que existan claves resultantes iguales para objetos diferentes, ya que el rango de posibles claves es mucho menor que el de posibles objetos a resumir (las claves suelen tener en torno al centenar de bits, pero los ficheros no tienen un tamaño límite).
- Son usadas en múltiples aplicaciones, como los arrays asociativos, criptografía, procesamiento de datos y firmas digitales, entre otros. Una buena función de hash es una que experimenta pocas colisiones en el conjunto esperado de entrada; es decir que se podrán identificar unívocamente las entradas.
- Las tablas hash, una de las aplicaciones más extendidas de las funciones de hash, aceleran el proceso de búsqueda de un registro de información según una clave (nota: este uso de la palabra poco se relaciona con su significado habitual). Por ejemplo, una cadena alfanumérica puede ser utilizada para buscar la información de un empleado en la base de datos de un sistema.
- La utilización de tablas hash provee un acceso casi directo a dichos registros, lo que significa que, en promedio, una búsqueda puede llegar a requerir sólo uno o dos intentos en la memoria o el archivo que contiene la información. Naturalmente, se prefiere una buena función de hash que evitará colisiones de hash.
- Si asumimos que la clave es una cadena de bytes, entonces la función de hash debería ser como un índice de los registros que tiene una distribución aleatoria sobre las cadenas de entrada esperadas. De otra forma, habría más colisiones de hash degradando así el tiempo de búsqueda. Si, por ejemplo, la clave es alfabética, cada byte puede tener sólo 26 de sus 256 valores posibles. Funciones tan simples no distribuirán los índices de una forma pareja.
- Para una comparación de velocidad y calidad de varias funciones de hash, referirse a los enlaces externos.

.5. Conclusión

El método de búsqueda hash o por transformación de clave aumenta la velocidad de búsqueda sin necesidad de que los elementos estén previamente ordenados.