Qué son las tablas hash

Una tabla Hash es un contenedor asociativo (tipo Diccionario) que permite un almacenamiento y posterior recuperación eficientes de elementos (denominados valores) a partir de otros objetos, llamados claves.

Una tabla hash se puede ver como un conjunto de entradas. Cada una de estas entradas tiene asociada una clave única, y por lo tanto, diferentes entradas de una misma tabla tendrán diferentes claves. Esto implica, que una clave identifica univocamente a una entrada en una tabla hash.

Por otro lado, las entradas de las tablas hash están compuestas por dos componentes, la propia clave y la información que se almacena en dicha entrada.

Estructura de una tabla hash

La estructura de las tablas hash es lo que les confiere su gran potencial, ya que hace de ellas unas estructuras extremadamente eficientes a la hora de recuperar información almacenada. El tiempo medio de recuperación de información es constante, es decir, no depende del tamaño de la tabla ni del número de elementos almacenados en la misma.

Una tabla hash está formada por un array de entradas, que será la estructura que almacene la información, y por una función de dispersión. La función de dispersión permite asociar el elemento almacenado en una entrada con la clave de dicha entrada. Por lo tanto, es un algoritmo crítico para el buen funcionamiento de la estructura.

Cuando se trabaja con tablas hash es frecuente que se produzcan colisiones. Las colisiones se producen cuando para dos elementos de información distintos, la función de dispersión les asigna la misma clave. Como se puede suponer, esta solución se debe arreglar de alguna forma. Para ello las tablas hash cuentan con una función de resolución de colisiones.

Existen dos tipos de tablas hash, en función de cómo resuelven las colisiones:

HASH (ABIERTO Y CERRADO)

La función de hash asigna un valor a una entrada dada de tal manera que el numero es ubicado en un array.

HASH CERRADO: vector que a medida que se le va introduciendo los números se va dispersando por el vector mediante una función ya preestablecida en el programa; cuando ocurre una colisión, es decir, que el numero que introdujo el usuario al aplicarle la función da una posición en el vector que ya estaba ocupada el programa aplicara una exploración para encontrar otra casilla desocupada.
Si en esta exploración no se encontró ninguna casilla disponible entonces quiere decir que el vector está lleno y termina el programa.

HASH ABIERTO: Es parecido al otro, la diferencia que tiene este con respecto al otro es que cuando colisiona coloca los números que colisiona en una lista, es decir, que trabaja el vector con memoria dinámica haciendo que la posición que colisiona el numero se va incrementado una lista de esa posición, por ejemplo:
Casillas
(0) -> 3->NULL
(1) -> 5->0->6->9->30->NULL
(2) ->NULL

- Una buena función hash debería satisfacer la suposición de hash uniforme.
- Como el recorrido de la función de hash es un número natural, hay que saber interpretar o transformar a número natural tipo de clave.
- Si se trata de claves enteras, el problema está más o menos resuelto.

Funciones Hash: Método de División

- **Método de división:**
- Este método consiste en tomar el resto de la división por m, el número de entradas de la tabla. Así
  h(k) = k mod m
  En C sería h(k) = k % m;
- Usar m = una potencia de 2 no es buena idea ya que el valor de hash queda dependiendo de sólo los bits menos significativos de k.
- Una forma de hacer hash(k) dependiente de todos los bits menos significativos es usar número primos no muy cercanos a una potencia de dos.

Operaciones básicas de las tablas hash

Insertar

El proceso de inserción en una tabla hash es muy simple y sencillo. Sobre el elemento que se desea insertar se aplica la función de dispersión. El valor obtenido tras la aplicación de esta función será el índice de la tabla en el que se insertará el nuevo elemento.

Borrar
El borrado en una tabla hash es muy sencillo y se realiza de forma muy eficiente. Una vez indicada la clave del objeto a borrar, se procederá a eliminar el valor asociado a dicha clave de la tabla.

Esta operación se realiza en tiempo constante, sin importar el tamaño de la tabla o el número de elementos que almacene en ese momento la estructura de datos. Esto es así ya que al ser la tabla una estructura a la que se puede acceder directamente a través de las claves, no es necesario recorrer toda la estructura para localizar un elemento determinado.

Si sobre la tabla resultante de la inserción del elemento azul realizamos el borrado del elemento negro, la tabla resultante sería la siguiente:

Otras operaciones:
Buscar un elemento en el hash
Obtener la clave donde esta un elemento dado

Ejemplo:

Dada la siguiente función de hash (K mod 5) ubicar los enteros siguientes.

23 34 54 12 43 76 34

Restos: 0, 1, 2 ,3 4

lista l;
l hash[5];

hash →

0	1	2	3	4
340	76	12	23 43	34 54

Definir las operaciones sobre hash: inicializar, insertar, buscar, eliminar, listar por casilla.

```
void inicializar (lista hash[], int largo){
  int i;
  for (i=0; i < largo; i++){
     hash[i] = Crearlista();
```

```
  }
}
```

Las restantes quedan como ejercicios