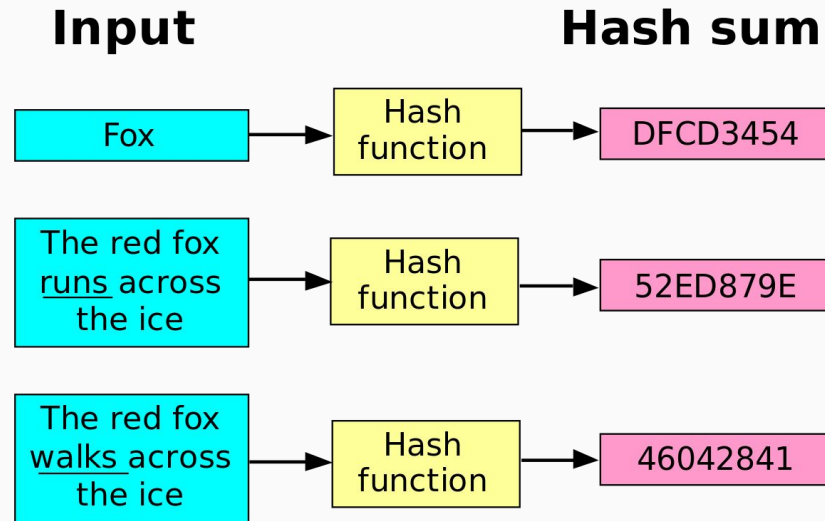


TUP - PROGRAMACION II

TABLAS HASH

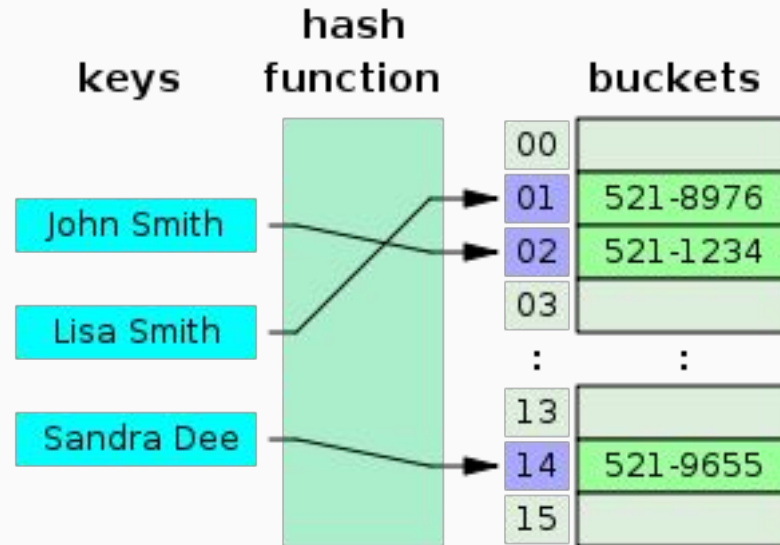
FUNCION HASH

- Una función hash es cualquier función que se puede utilizar para asignar datos de tamaño arbitrario a datos de tamaño fijo.
- Los valores devueltos por una función hash se denominan valores hash, códigos hash, digest, o simplemente hashes.



TABLAS HASH

- Es una estructura de datos que asocia llaves o claves con valores.
- Funciona transformando la clave con una función hash en un hash, un número que identifica la posición (casilla o campo) donde la tabla hash localiza el valor deseado.
- La operación principal que soporta de manera eficiente es la búsqueda.



TABLAS HASH - Inserción

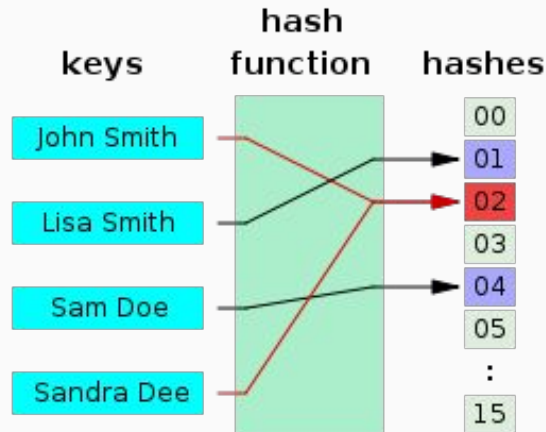
- Para almacenar un elemento en la tabla hash se ha de convertir su clave a un número. Esto se consigue aplicando la función hash a la clave del elemento.
- El resultado de la función resumen ha de mapearse al espacio de direcciones del vector que se emplea como soporte. Tras este paso se obtiene un índice válido para la tabla.
- El elemento se almacena en la posición de la tabla obtenido en el paso anterior.
 - Si en la posición de la tabla ya había otro elemento, se ha producido una colisión. Este problema se puede solucionar siguiendo distintas técnicas.

TABLAS HASH - Búsqueda

- Para recuperar los datos, es necesario únicamente conocer la clave del elemento, a la cual se le aplica la función hash.
- El valor obtenido se mapea al espacio de direcciones de la tabla.
- Si el elemento existente en la posición indicada en el paso anterior tiene la misma clave que la empleada en la búsqueda, entonces es el deseado. Si la clave es distinta, se ha de buscar el elemento según la técnica empleada para resolver el problema de las colisiones al almacenar el elemento.

TABLAS HASH - Colisiones

- Si dos llaves generan un hash apuntando al mismo índice, los registros correspondientes no pueden ser almacenados en la misma posición.
- Cuando una casilla ya está ocupada, debemos encontrar otra ubicación donde almacenar el nuevo registro, y hacerlo de tal manera que podamos encontrarlo cuando se requiera.



TABLAS HASH - Direcccionamiento Cerrado, Encadenamiento separado o Hashing abierto

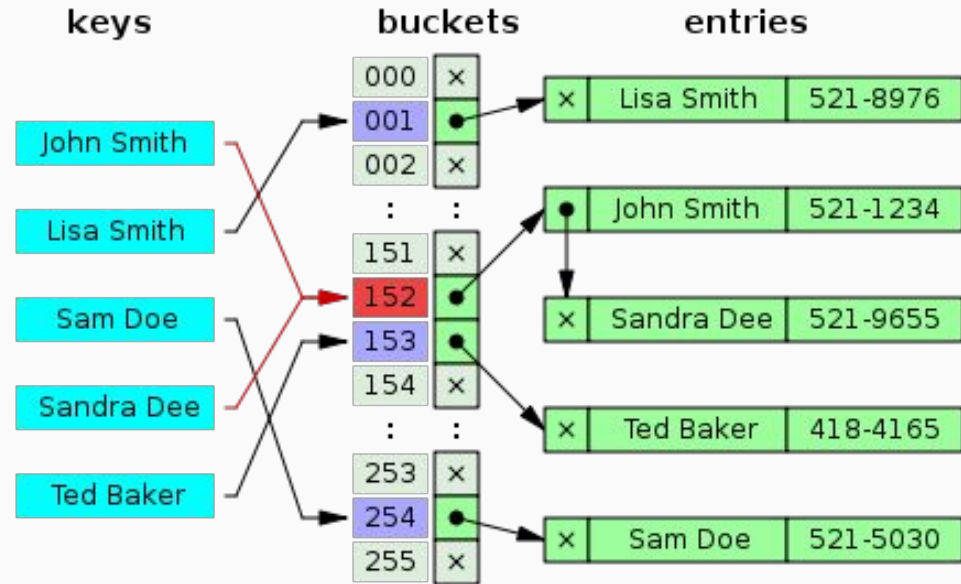
- Cada casilla en el array referencia una lista de los registros insertados que colisionan en la misma casilla. La inserción consiste en encontrar la casilla correcta y agregar al final de la lista correspondiente. El borrado consiste en buscar y quitar de la lista.

Técnicas:

- **Encadenamiento separado con Listas Enlazadas:** El Array de índices hash posee un puntero a una lista enlazada que es donde se almacenan los valores.
- **Encadenamiento separado por Cabecera de Celdas:** Almacena el valor del primer registro directamente junto a la clave Hash y luego utiliza una lista para almacenar el resto de los valores.
- **Encadenamiento separado por otras estructuras:** Las listas pueden ser reemplazadas por árboles auto-balanceables, por ejemplo, el tiempo teórico del peor de los casos disminuye de $O(n)$ a $O(\log n)$.

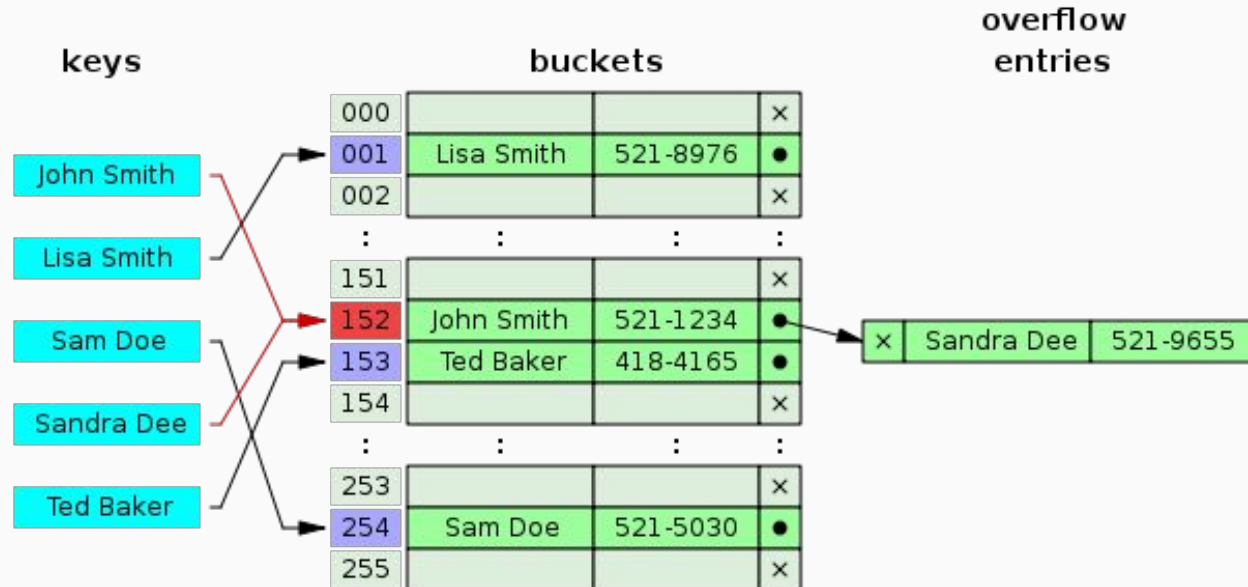
TABLAS HASH - Encadenamiento separado con Listas Enlazadas

- **Encadenamiento separado con Listas Enlazadas:** El Array de índices hash posee un puntero a una lista enlazada que es donde se almacenan los valores.



TABLAS HASH - Encadenamiento separado con Cabecera de Celdas

- **Encadenamiento separado por Cabecera de Celdas:** Almacena el valor del primer registro directamente junto a la clave Hash y luego utiliza una lista para almacenar el resto de los valores.



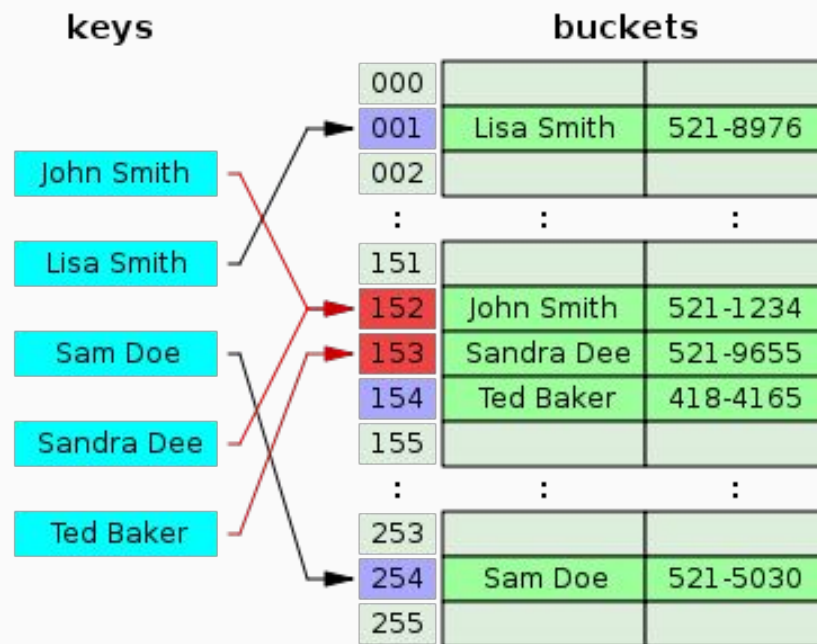
TABLAS HASH - Direcccionamiento abierto o Hashing cerrado

- Las tablas hash de direccionamiento abierto pueden almacenar los registros directamente en el array.
- Las colisiones se resuelven mediante un sondeo del array, en el que se buscan diferentes posiciones del array (secuencia de sondeo) hasta que el registro es encontrado o se llega a una casilla vacía, indicando que no existe esa llave en la tabla.

Las secuencias de sondeo más socorridas incluyen:

- **Sondeo lineal:** en el que el intervalo entre cada intento es constante (frecuentemente 1).
- **Sondeo cuadrático:** en el que el intervalo entre los intentos aumenta linealmente (por lo que los índices son descritos por una función cuadrática).
- **Sondeo hashe:** en el que el intervalo entre intentos es constante para cada registro pero es calculado por otra función hash.

TABLAS HASH - Direccionamiento abierto o Hashing cerrado



TABLAS HASH - Implementación

```
registro par { llave, valor }  
var vector de pares casilla[0..numcasillas-1]
```

```
function buscacasilla(llave) {  
    i := hash(llave) módulo de numcasillas  
    loop {  
        if casilla[i] esta libre or casilla[i].llave = llave  
            return i  
        i := (i + 1) módulo de numcasillas  
    }  
}
```

```
function busqueda(llave)  
    i := buscacasilla(llave)  
    if casilla[i] está ocupada // llave está en la tabla  
        return casilla[i].valor  
    else // llave no está en la tabla  
        return no encontrada
```

```
function asignar(llave, valor) {  
    i := buscacasilla(llave)  
    if casilla[i] está ocupada  
        casilla[i].valor := valor  
    else {  
        if tabla casi llena {  
            hacer tabla más grande (nota 1)  
            i := buscacasilla(llave)  
        }  
        casilla[i].llave := llave  
        casilla[i].valor := valor  
    }  
}
```