Tablas de Dispersión

TDA - Tabla

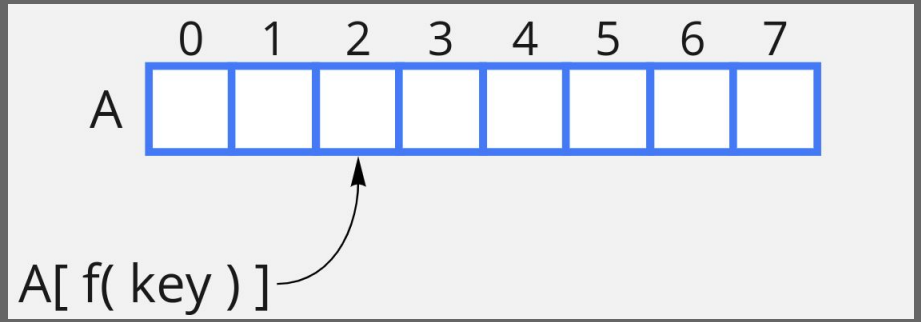
Dado un conjunto de índices o llaves K y un conjunto de valores V, se tiene una función $f: K \rightarrow V$ tal que permite realizar las siguientes operaciones:

- Obtener el valor guardado en la tabla para el índice **i**.
- Modificar el valor guardado en la tabla para el índice **i**.

ED
ESTRUCTURAS DE DATOS

Función de dispersión (Hashing function)

Una función de dispersión **f** es una función que recibe un atributo o conjunto de atributos de un elemento, al que se llamará *llave* o *key*, tal que f(*key*) ϵ $S \subseteq \mathbb{N}$.



Al crear una función de dispersión deben seguirse tres reglas:

1. Debe ser sencilla de computar.
2. Debe dispersar los datos de manera uniforme.
3. Debe ser consistente.

ED
ESTRUCTURAS DE DATOS

Métodos comunes en funciones de dispersión

Truncamiento: Toma solo parte de la llave para el procesamiento.

Ejemplo:

Sea A un arreglo de 100 entradas y *key* una llave numérica, la función solo devolverá los últimos dos dígitos.

f(12345) = 45
f(3) = 3

Folding: Parte la llave en subpartes y luego las mezcla para obtener un resultado.

Ejemplo:

Sea A un arreglo de 100 entradas y *key* una llave numérica, la función dividirá la llave en grupos de 2 dígitos de derecha a izquierda y luego los sumará.

$$f(12345) = 1 + 23 + 45 = 69$$
$$f(2468) = 24 + 68 = 92$$

Módulo aritmético: Si la llave es numérica, la dividimos entre el tamaño del arreglo y tomamos el residuo.

Ejemplo:

Sea A un arreglo de 20 entradas y *key* una llave numérica.

f(93) = 93 % 20 = 13
f(12) = 12 % 20 = 12
f(100) = 100 % 20 = 0

ED
ESTRUCTURAS DE DATOS

Método hashCode en Java

En Java, la clase Object tiene el método

public int hashCode()

A tomar en cuenta:

- Los resultados deben ser consistentes durante la ejecución del programa.
- Dos objetos iguales bajo el criterio de la función *equals()* deben devolver el mismo hashCode.

Tablas de dispersión (Hash tables) / Diccionarios

Una tabla que usa una función de dispersión para guardar sus elementos es una tabla de dispersión.

ED
ESTRUCTURAS DE DATOS

Factor de carga

Si el número de localidades con los que cuenta la tabla es **m** y la cantidad actual de elementos guardados en ella es **n**, entonces el factor de carga se define como $\alpha = n / m$.

ED
ESTRUCTURAS DE DATOS

Colisiones

Sea f la función de dispersión de la tabla, y sean dos elementos e_1 y e_2 tal que e_1.key ≠ e_2.key:

si f(e_1.key) = f(e_2.key) se dice que ocurrió una colisión.

ED
ESTRUCTURAS DE DATOS

Resolución de colisiones

Dispersión perfecta: Se tiene una función de dispersión sin colisiones.

- Se tienen que conocer a detalle los elementos que se guardarán en la tabla.
- El número de localidades de la tabla debe ser el número total de posibles valores a guardar.

ED
ESTRUCTURAS DE DATOS

Direccionamiento abierto: Se busca una localidad vacía en caso de colisión.

- **Exploración lineal:** Si la localidad **h** no está disponible, intentará con **h** + 1, **h** + 2, **h** + 3, ... , **h** + **i**.

- **Exploración cuadrática:** Si la localidad **h** no está disponible, intentará con **h** + 1, **h** + 4, **h** + 9, ... , **h** + i^2.

ED
ESTRUCTURAS DE DATOS

- **Exploración aleatoria**: Se usa una función auxiliar que devuelve un número aleatorio que denotará el salto de localidades. Debe usarse la misma semilla para generar los mismos números aleatorios cada vez.

Dada la siguiente secuencia de números aleatorios
S: 4, 1, 9, 5, 2, 4, 7, 5, 6

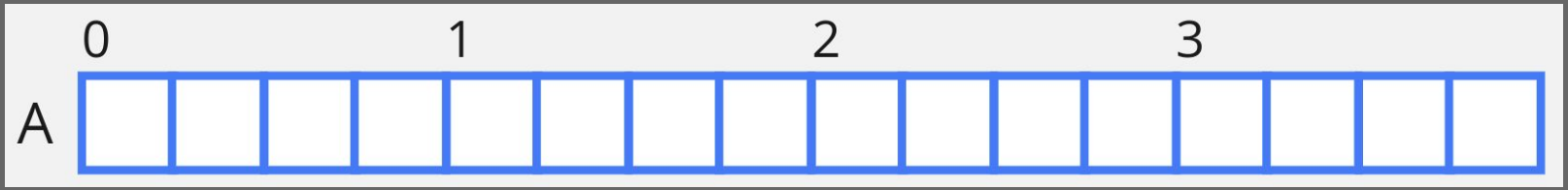
$$h_1 = h_0 + 4$$
$$h_2 = h_1 + 1$$
$$h_3 = h_2 + 9$$
$$\ldots$$

Al hacer cualquier tipo de exploración, es importante aplicar módulo para que el nuevo valor no se salga del límite de localidades.

ED
ESTRUCTURAS DE DATOS

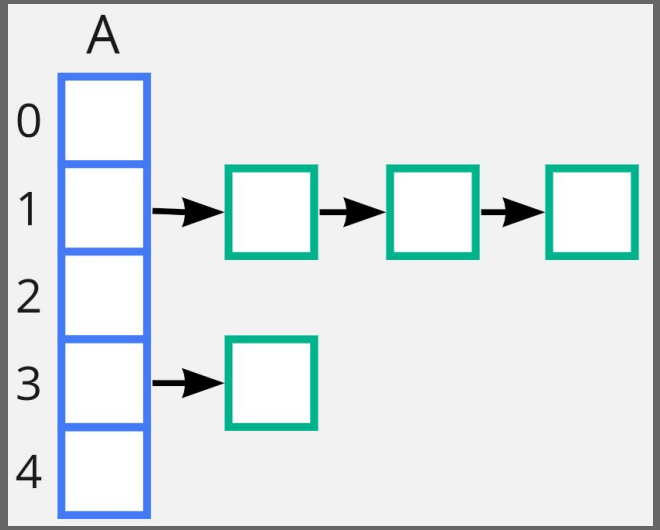
- **Doble dispersión**: En caso de colisión con la función **f**, se otra función de dispersión **g** que encontrará una nueva localidad.

- **Uso de cubetas**: Para cada llave hay cierta cantidad de localidades disponibles.



ED
ESTRUCTURAS DE DATOS

Encadenamiento: En cada localidad se tiene una estructura de datos auxiliar.



Usando exploración abierta o encadenamiento se recomienda que la cantidad de localidades en la tabla sea un número primo para reducir el número de colisiones.

ED
ESTRUCTURAS DE DATOS

Análisis de tiempo

	Búsqueda fallida	Búsqueda exitosa
Dispersión perfecta	$O(1)$	$O(1)$
Direccionamiento abierto con $\alpha < 1$	$O(1 / (1 - \alpha))$	$O(\ln(1 / (1 - \alpha)) / \alpha)$
Encadenamiento	$O(1 + \alpha)$	$O(1 + \alpha)$

Se asume el uso de una buena función de dispersión.

ED
ESTRUCTURAS DE DATOS

- Simulador de redireccionamiento abierto con exploraciones.
- Simulador de redireccionamiento abierto con cubetas.
- Simulador de encadenamiento con listas.