

Hashing

Ing. Armandina Leal Flores

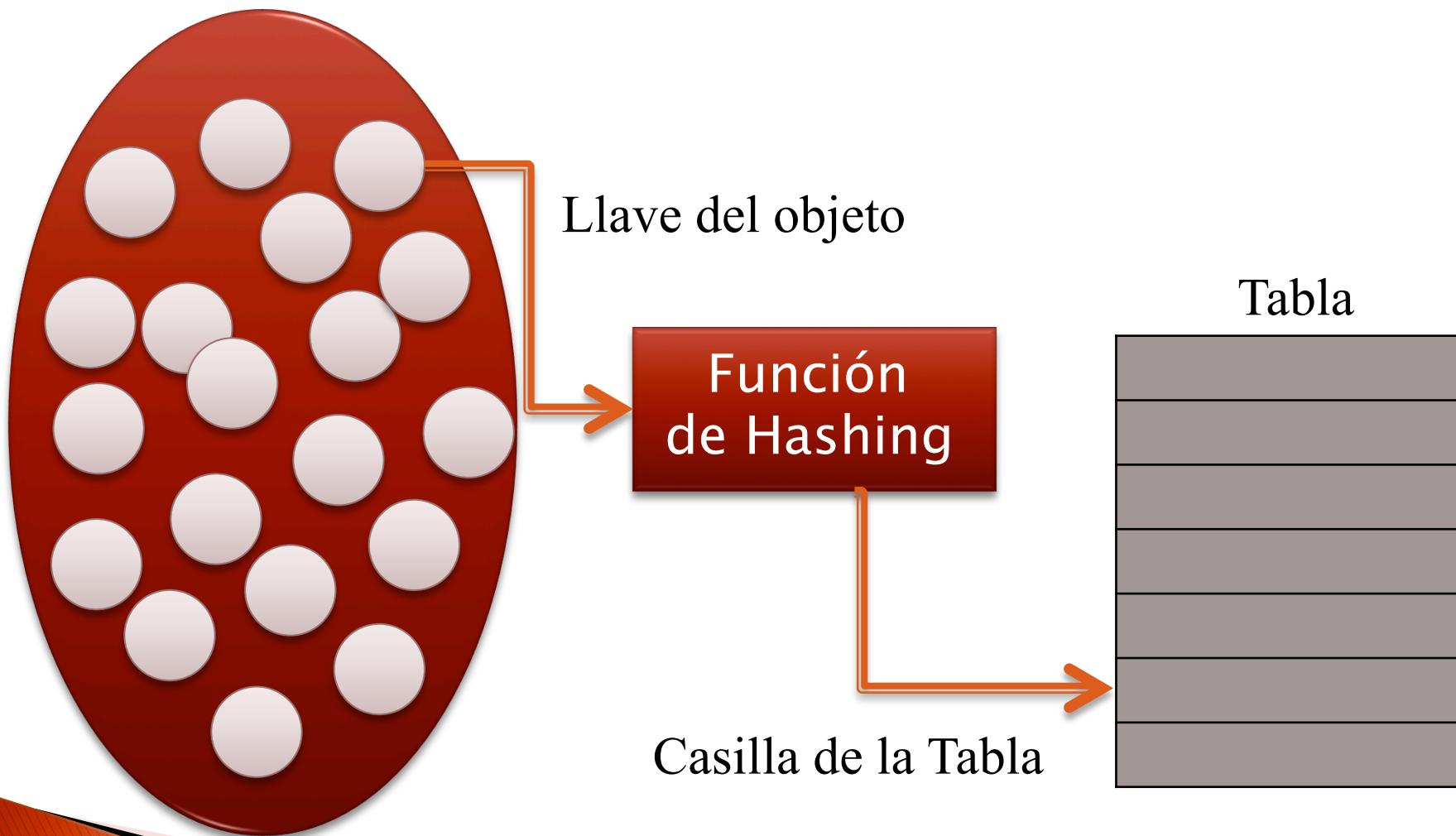
Estructura de datos **Conjunto**

- ▶ Permite almacenar un grupo de datos que su única relación es pertenecer al conjunto.
- ▶ El conjunto se emplea para la búsqueda de información utilizando la propiedad de **PERTENENCIA**.
- ▶ Todos los elementos del conjunto deben ser diferentes.

Implementación de un Conjunto

- ▶ Para implementar un conjunto se utiliza un arreglo unidimensional de objetos al que se denomina **TABLA**.
- ▶ Los elementos de una tabla se distinguen porque tienen una **LLAVE** que los identifica en forma única.
 - Ejemplo de llaves:
Número de Matrícula, Número de Nómina, CURP, Registro Federal de contribuyentes, etc.
- ▶ El orden en que se almacenan los elementos en la Tabla se determina empleando la **Técnica de HASHING**.

Técnica de HASHING...



Técnica de Hashing

- ▶ Pasos para implementar la Técnica
 1. Diseñar función de Hashing
 2. Implementar la **función de Hashing**
 3. Definir la estrategia para controlar las **COLISIONES**.

*Una **colisión** ocurre cuando la función de Hash le asigna a dos objetos la misma casilla del arreglo.*

Función de Hashing

- ▶ Debe tener por lo menos las siguientes características:
 - Se ejecuta rápidamente.
 - Distribuye los objetos en toda la tabla.
 - Minimiza la cantidad de colisiones.
 - Toma en cuenta el tamaño de la tabla.
- ▶ La función supone que la llave es numérica. Si la llave es alfabética se traduce empleando el código ASCII.
- ▶ Algunos métodos son:
 - Selección de Dígitos.
 - Residuales (División).
 - Folding (Plegamiento).

Selección de Dígitos

- ▶ Selecciona algunos dígitos de la llave y con ellos generar el índice para la Tabla.

- ▶ Ejemplo:

$$H(d_1d_2d_3d_4d_5d_6) = d_2d_4d_6 \text{ donde } 0 \leq d_i \leq 9$$

- ▶ Esto serviría para acceder a una Tabla de 1000 elementos (000-999)

Residuales

- ▶ Es una de los métodos más utilizados.
- ▶ Al objeto le asigna la casilla que resulte del residuo de dividir la llave entre el tamaño de la Tabla.
- ▶ Ejemplo:

$$H(d_1 d_2 d_3 d_4 d_5 d_6) = d_1 d_2 d_3 d_4 d_5 d_6 \% N$$

donde $0 \leq H(\text{llave}) \leq N-1$

N debe ser
primo

Folding

- ▶ Es una técnica muy utilizada.
- ▶ Consiste en aplicar operaciones sobre algunos de los dígitos de la llave
- ▶ Ejemplo:

$$H(d_1d_2d_3d_4d_5d_6) = d_1 + d_2 + d_3 + d_4 + d_5 + d_6$$

donde $0 \leq d_i \leq 9$ y $0 \leq H(\text{llave}) \leq 54$

Ejemplos...

- ▶ Suponer que la llave es la matrícula de un estudiante del Tec
 - Llave = $d_1d_2d_3d_4d_5d_6$
- ▶ Si la tabla es de 100 elementos ¿qué tan recomendable es seleccionar a los primeros 2 dígitos de la matrícula como índice para la Tabla?
- ▶ NO ES RECOMENDABLE (demasiadas repeticiones).
- ▶ Si la tabla es de 100 elementos ¿qué tan recomendable sería sumar todos los dígitos de la matrícula y usar el resultado como índice para la Tabla?
- ▶ NO ES RECOMENDABLE (el mayor índice sería 54)

Estrategias para el manejo de Colisiones

- ▶ Métodos de Dirección Abierta
(métodos de Hashing Cerrado o de Rehashing).
- ▶ Métodos de Encadenamiento
(métodos de Hashing Abierto).

Métodos de Dirección Abierta.

- ▶ Los elementos que colisionan se acomodan dentro de la misma tabla pero en una casilla diferente.
- ▶ Para maximizar el uso del espacio, la Tabla se maneja como una arreglo CIRCULAR.

Métodos de Dirección Abierta

- ▶ Los principales métodos de la categoría de Dirección Abierta son:
 - Prueba Lineal
 - Prueba Cuadrática
 - Prueba Reasignación Aleatoria
 - Prueba Doble Hashing

Prueba Lineal

- ▶ Cuando un elemento colisiona, se busca en forma secuencial una posición disponible, a partir de la dirección base del elemento.

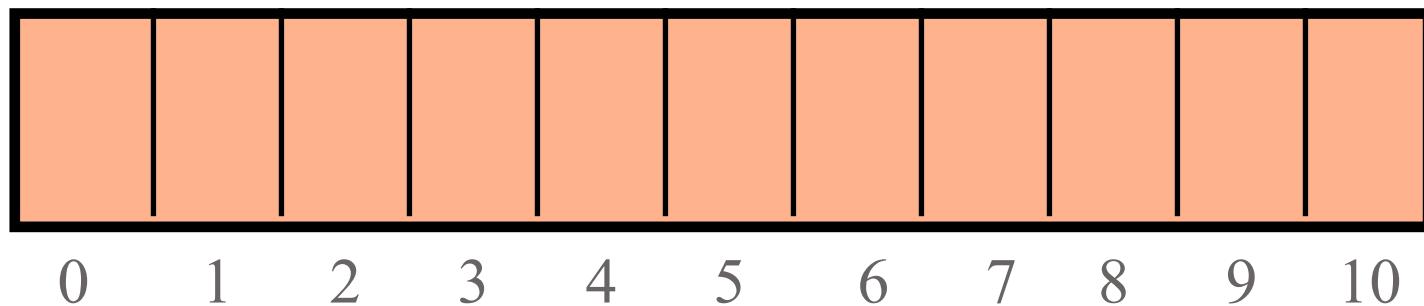
$$\text{Nueva_posición} = \text{dir_base} + J$$

donde J es la cantidad de colisiones que ha provocado esa llave.

- ▶ Es la técnica sencilla de implementar, pero es muy ineficiente

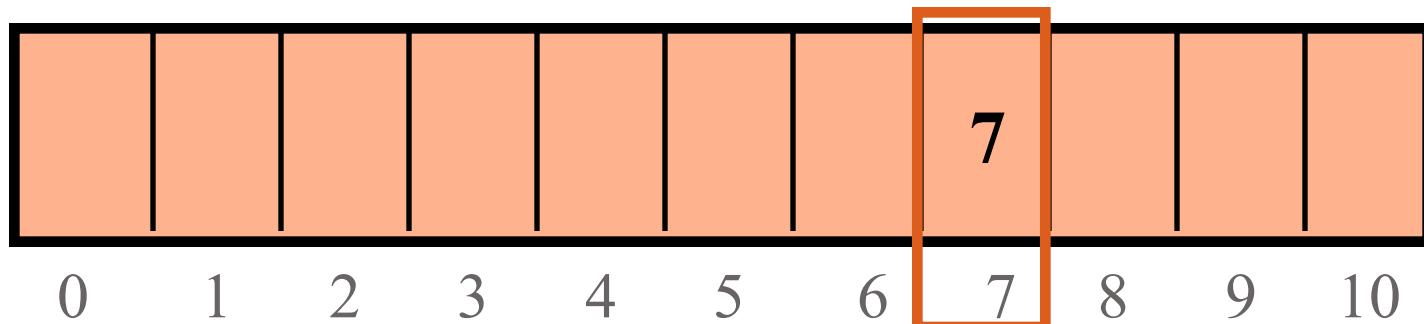
Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \%} 11$
- ▶ Insertar la secuencia: 7, 17, 6, 33, 28, 52



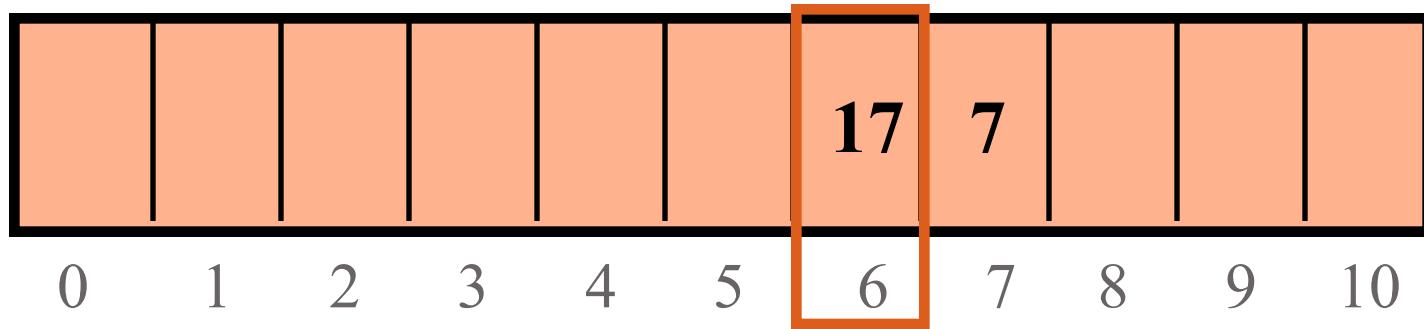
Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: 7, 17, 6, 33, 28, 52



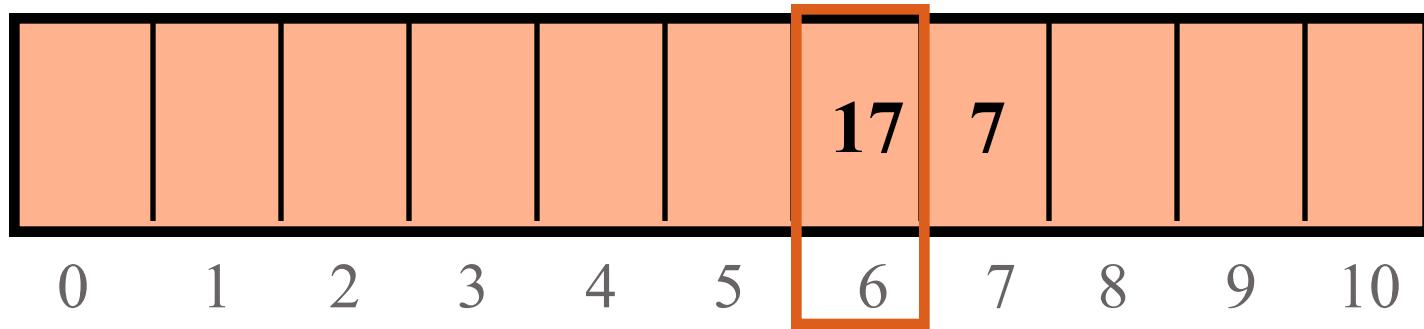
Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: **7, 17, 6, 33, 28, 52**



Ejemplo Prueba Lineal...

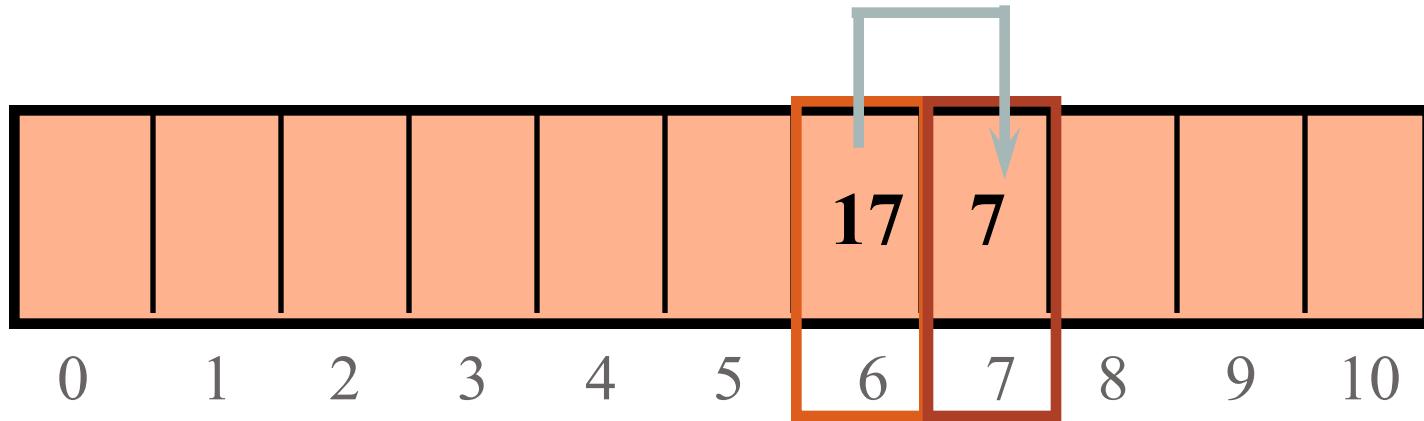
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: 7, 17, 6, 33, 28, 52



COLISIÓN

Ejemplo Prueba Lineal...

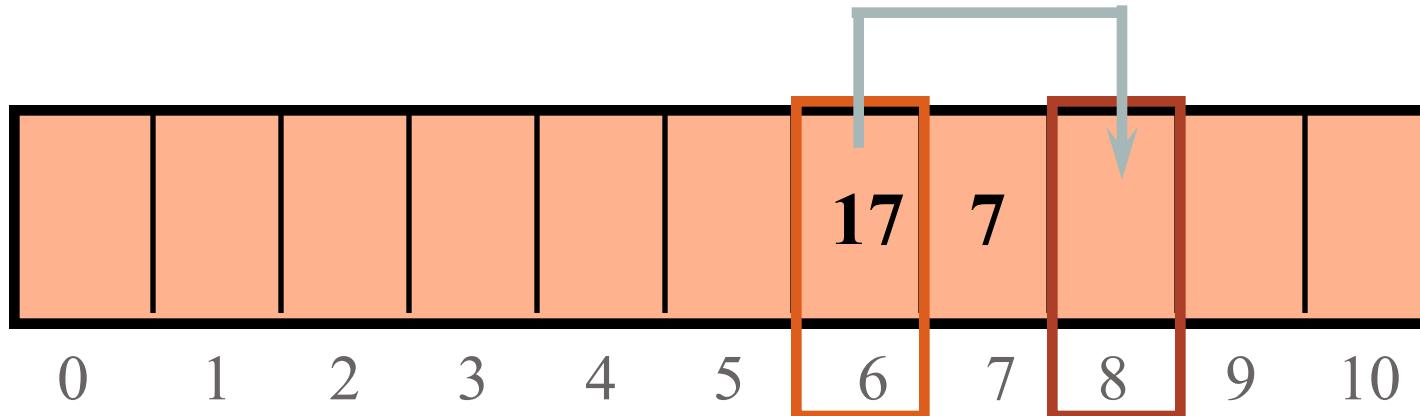
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: **7, 17, 6, 33, 28, 52**



COLISIÓN

Ejemplo Prueba Lineal...

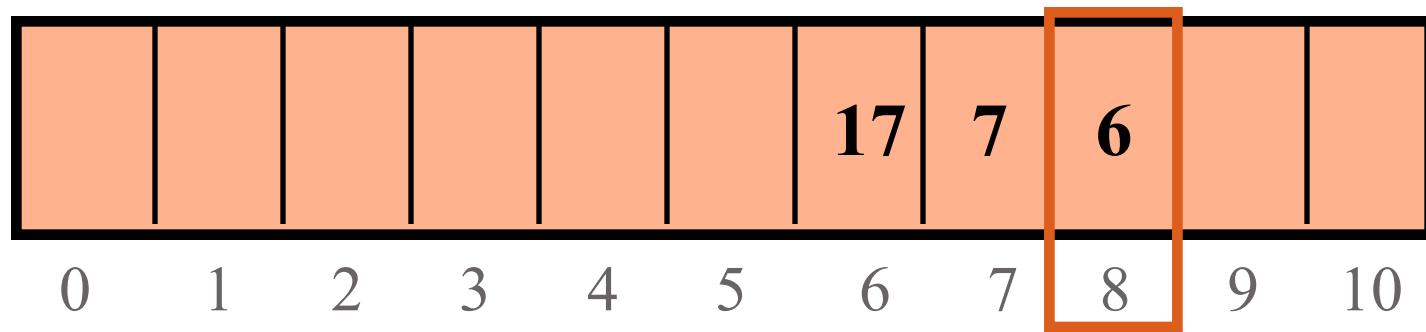
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: **7, 17, 6, 33, 28, 52**



LUGAR DISPONIBLE

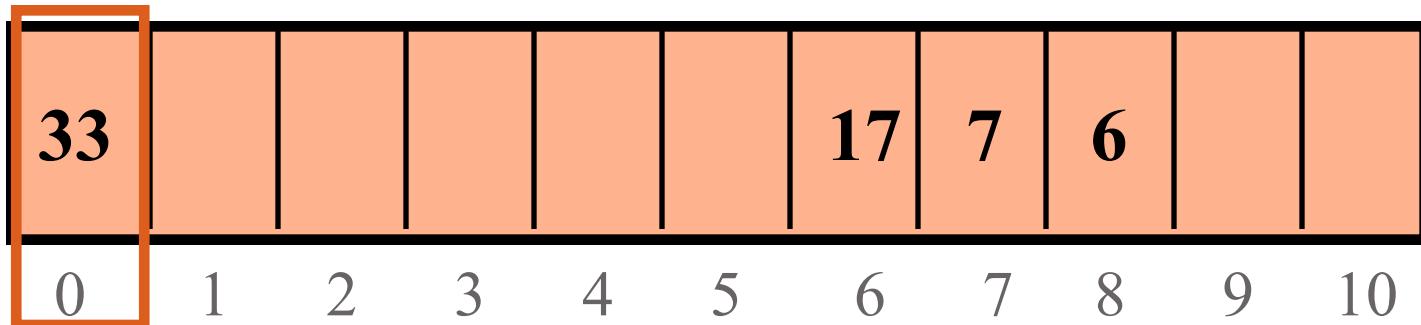
Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: **7, 17, 6, 33, 28, 52**



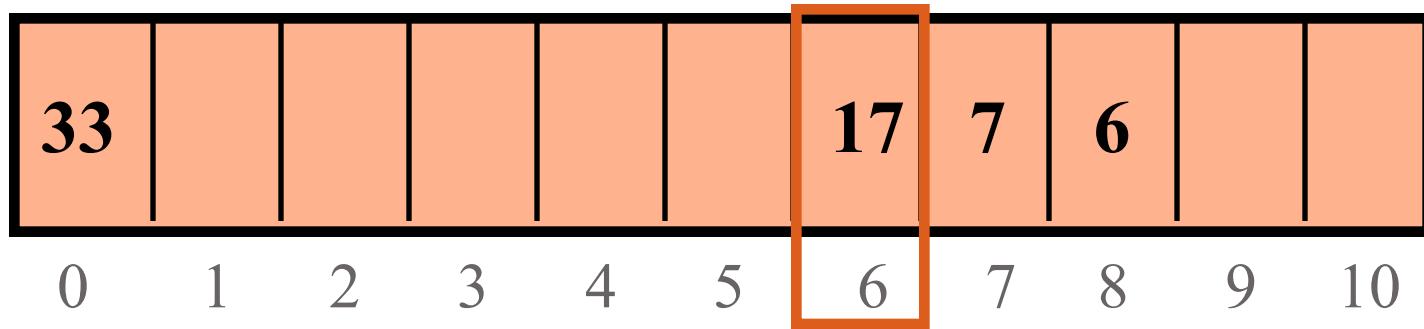
Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: **7, 17, 6, 33, 28, 52**



Ejemplo Prueba Lineal...

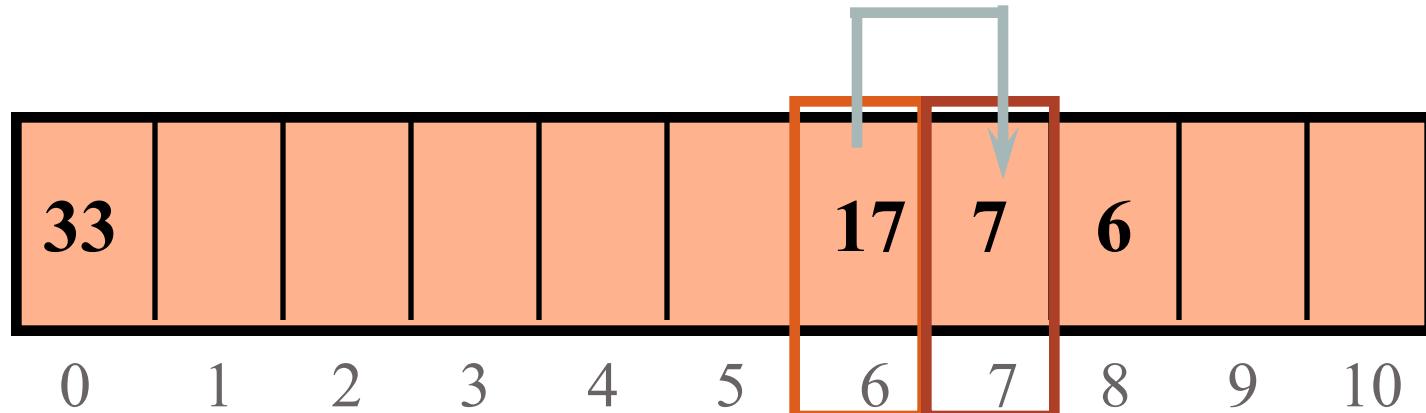
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: **7, 17, 6, 33, 28, 52**



COLISIÓN

Ejemplo Prueba Lineal...

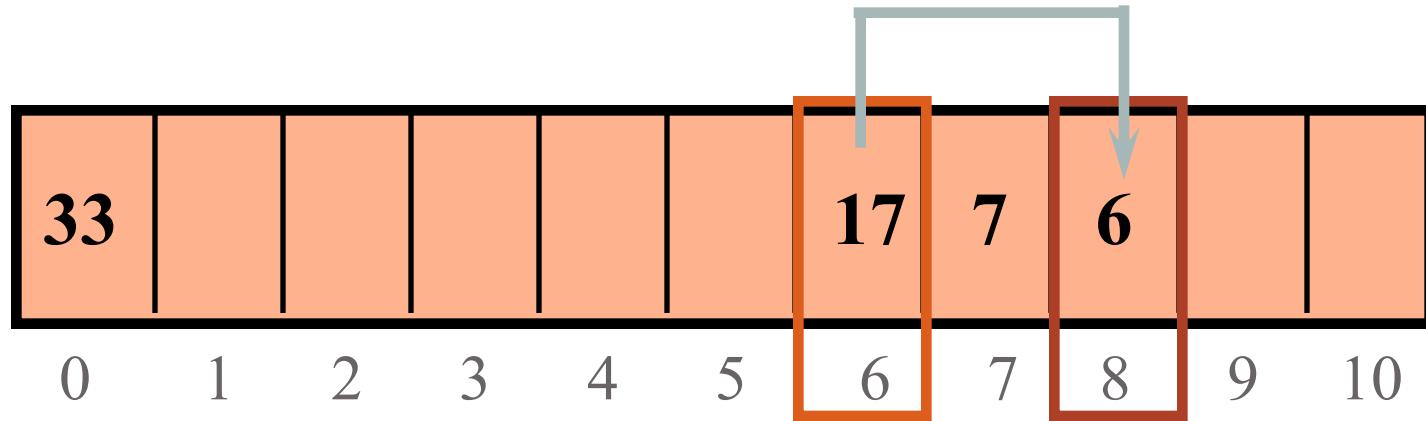
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: **7, 17, 6, 33, 28, 52**



COLISIÓN

Ejemplo Prueba Lineal...

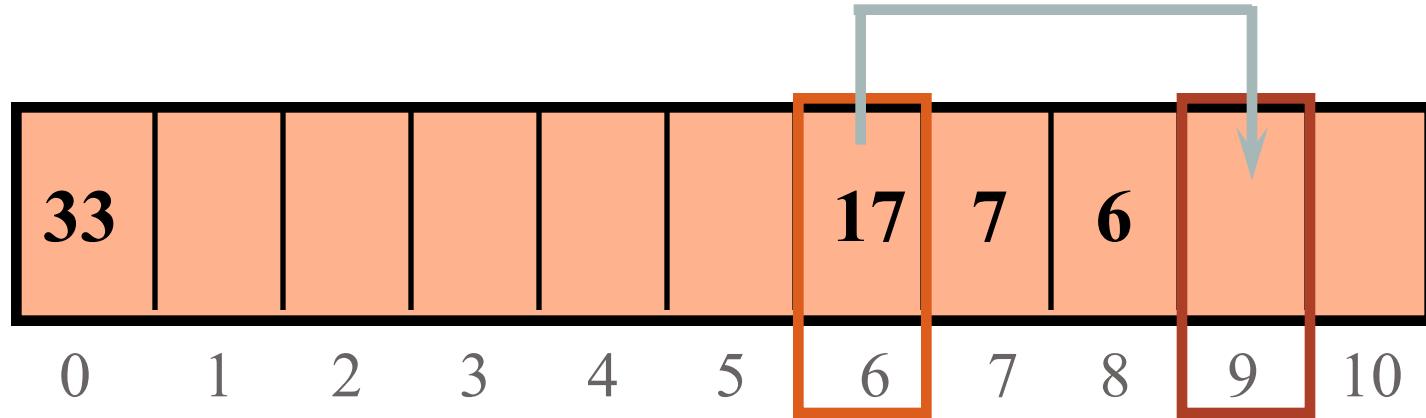
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: **7, 17, 6, 33, 28, 52**



COLISIÓN

Ejemplo Prueba Lineal...

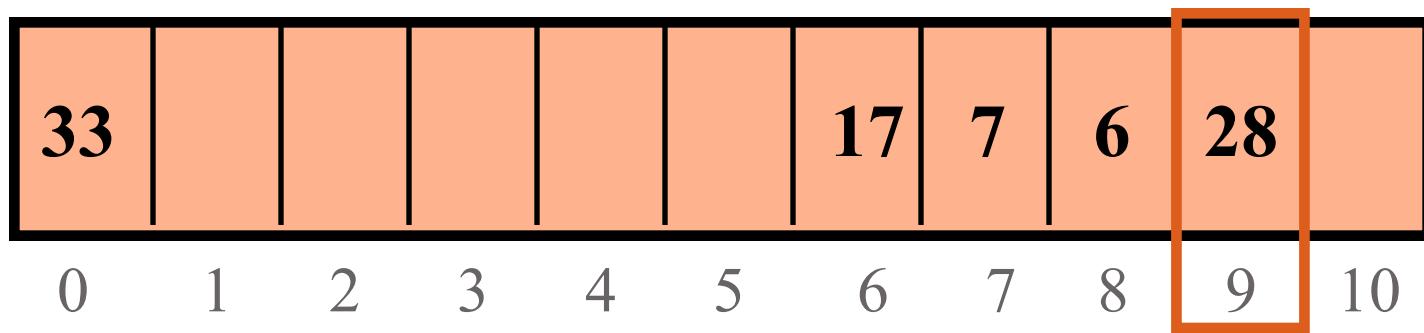
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: **7, 17, 6, 33, 28, 52**



LUGAR DISPONIBLE

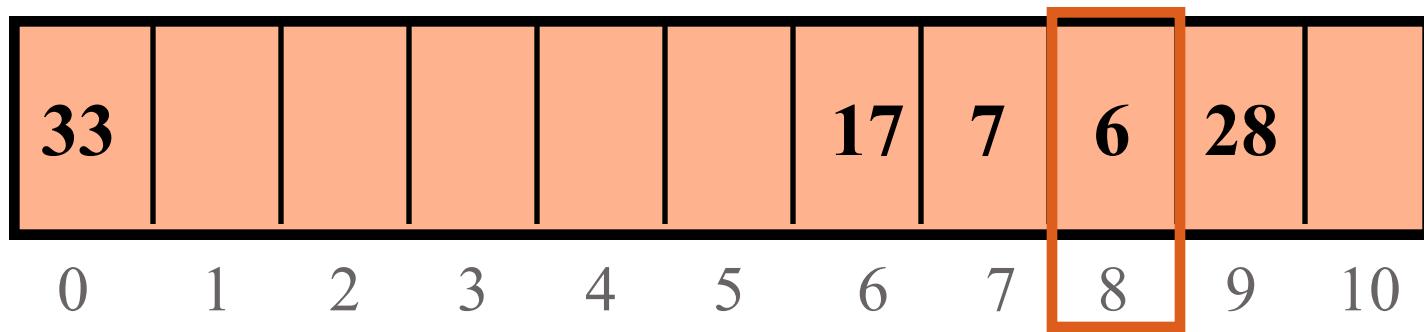
Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: **7, 17, 6, 33, 28, 52**



Ejemplo Prueba Lineal...

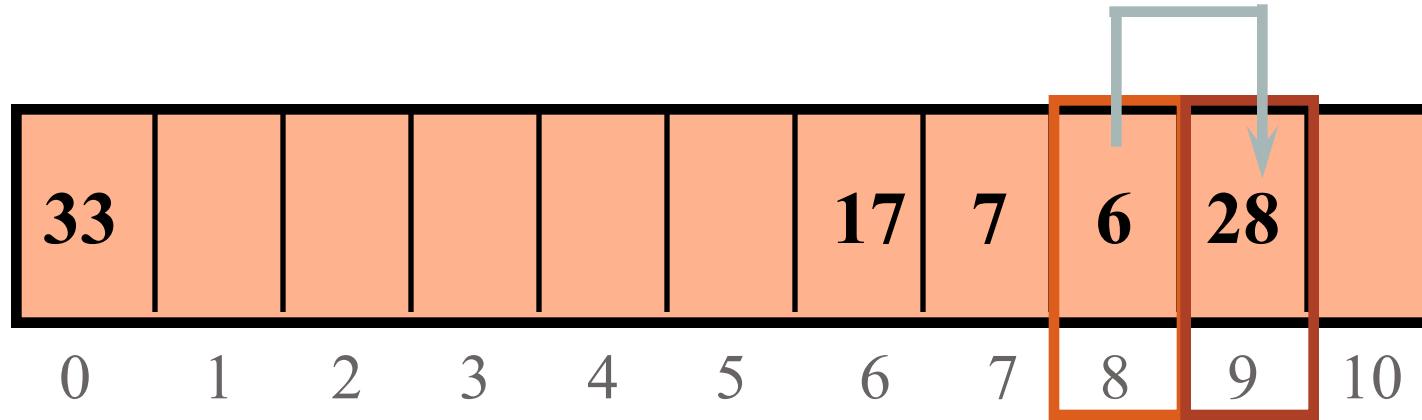
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: **7, 17, 6, 33, 28, 52**



COLISIÓN

Ejemplo Prueba Lineal...

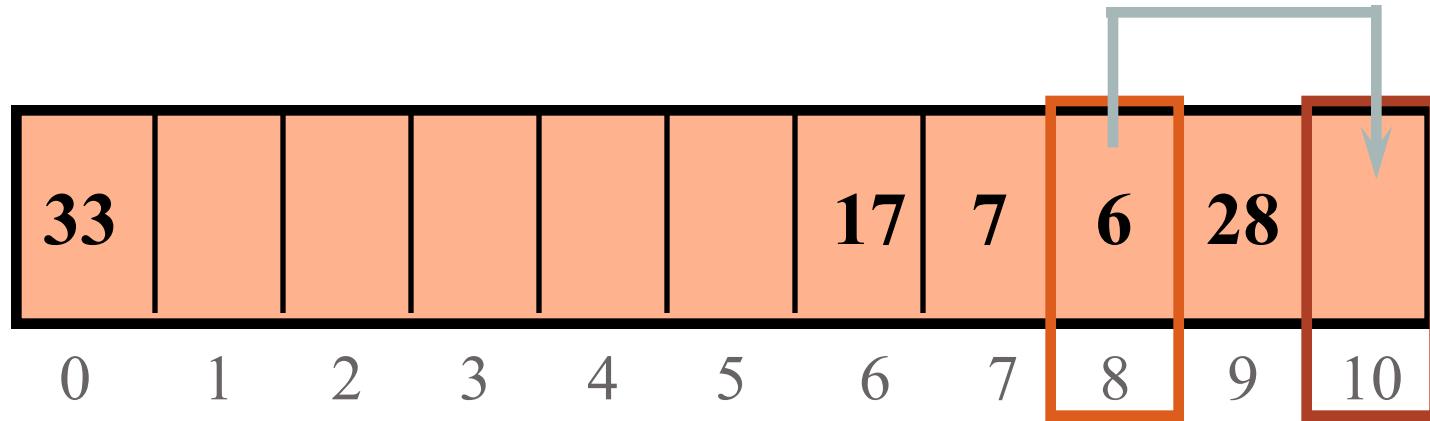
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: **7, 17, 6, 33, 28, 52**



COLISIÓN

Ejemplo Prueba Lineal...

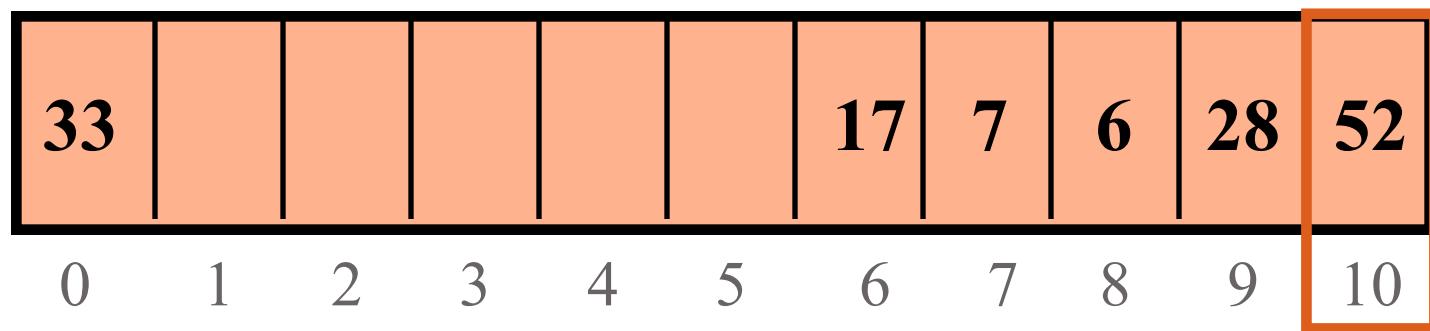
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: **7, 17, 6, 33, 28, 52**



LUGAR DISPONIBLE

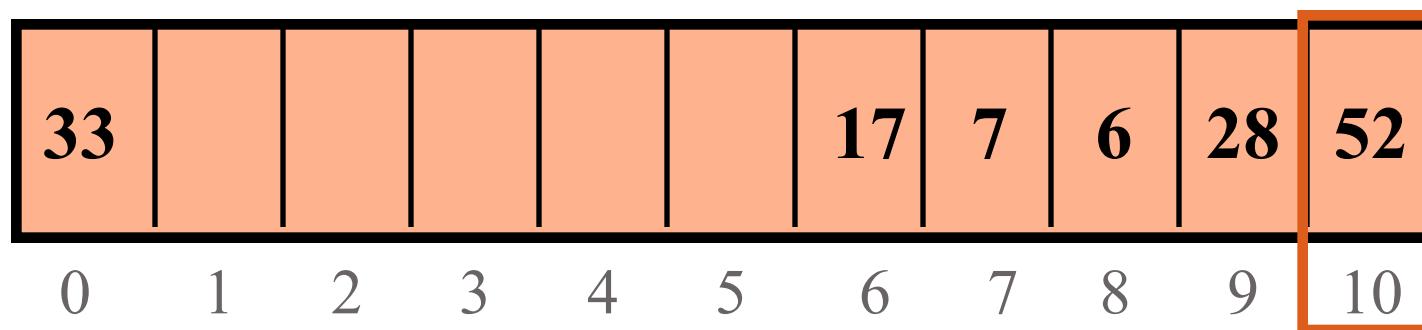
Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Insertar la secuencia: **7, 17, 6, 33, 28, 52**



Ejemplo Prueba Lineal...

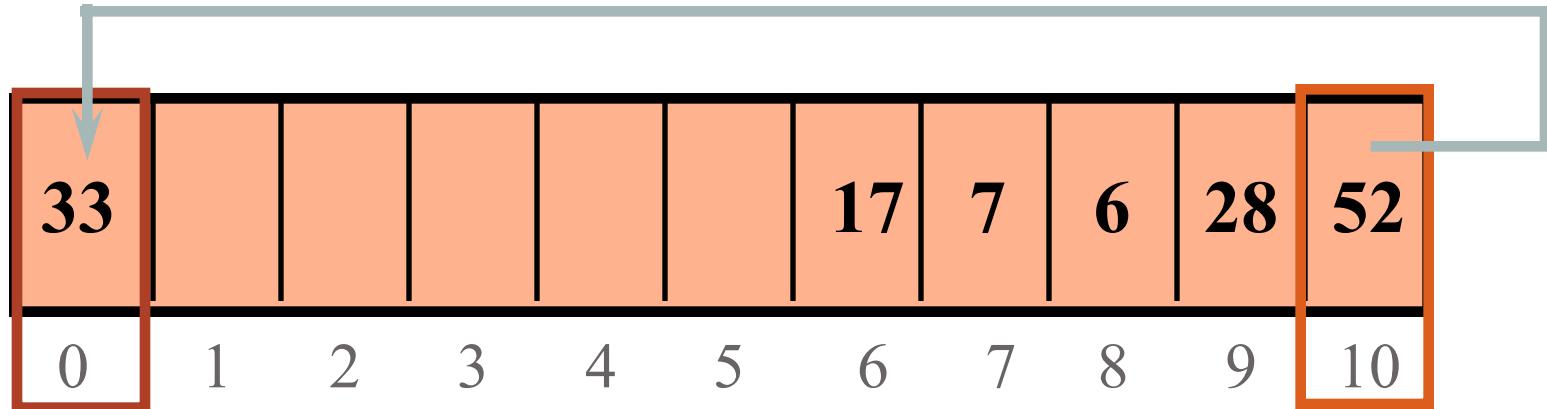
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Secuencia: 7, 17, 6, 33, 28, 52
- ▶ Agregar el 76



COLISIÓN

Ejemplo Prueba Lineal...

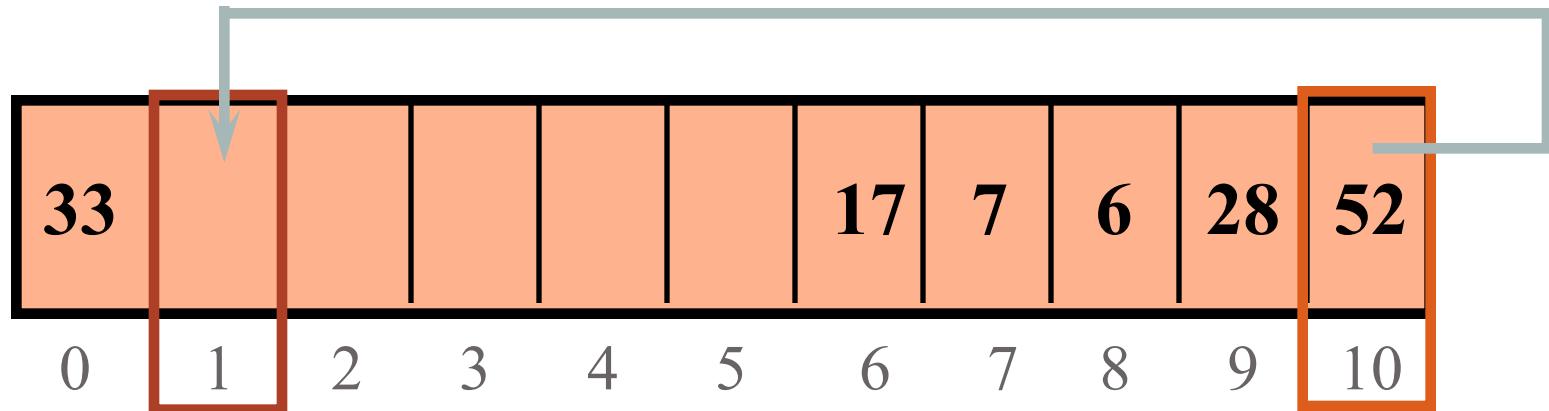
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Secuencia: 7, 17, 6, 33, 28, 52
- ▶ Agregar el 76



COLISIÓN

Ejemplo Prueba Lineal...

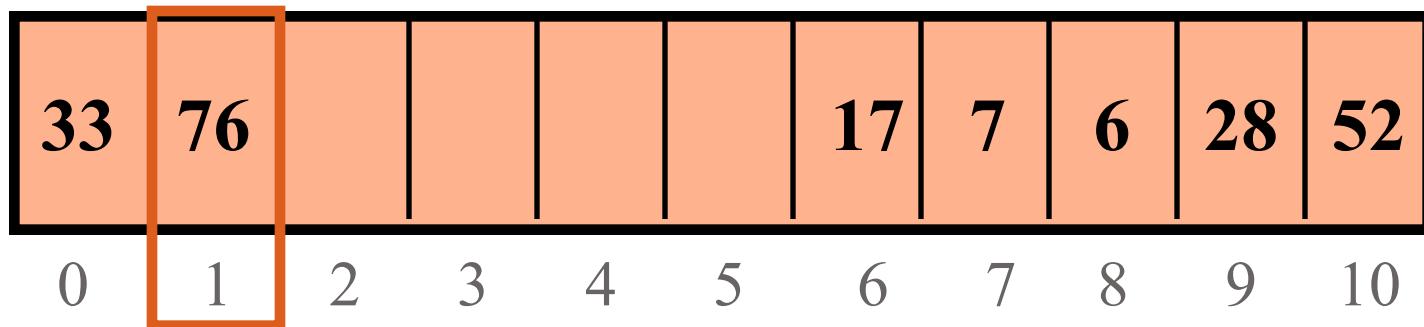
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Secuencia: 7, 17, 6, 33, 28, 52
- ▶ Agregar el 76



LUGAR DISPONIBLE

Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Secuencia: 7, 17, 6, 33, 28, 52
- ▶ Agregar el 76



Prueba Lineal

► Búsqueda de un elemento:

1. Utilizar la función de Hashing para calcular la dirección base del elemento buscado.
2. Repetir:

Compara el contenido de la casilla con el elemento.

Si es el elemento buscado : *EXITO!!* y termina.

Si no.. se mueve a la siguiente posición hasta que:

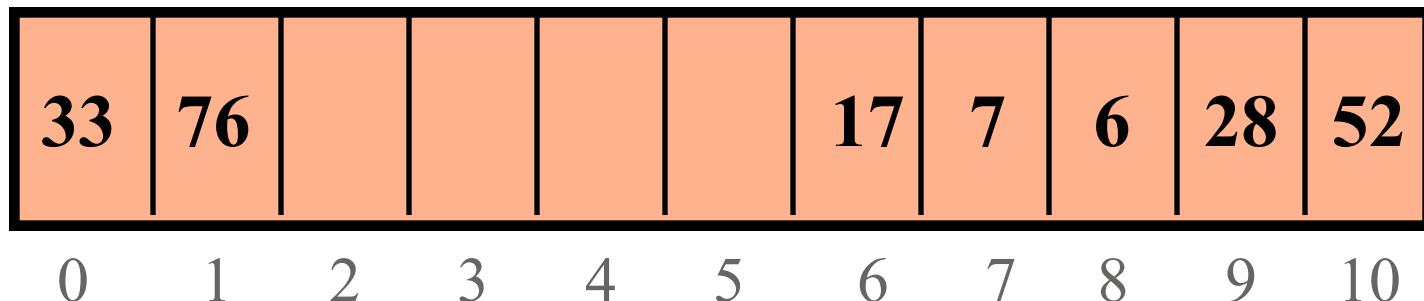
- a) Se encuentre el elemento o
- b) Se llegue a una casilla vacía o
- c) Se llegue a la dirección base.

Si se llega a una posición vacía o la dirección base, *NO SE ENCONTRÓ EL ELEMENTO.*

► Los procesos de Inserción y Eliminación llevan implícito al proceso de búsqueda.

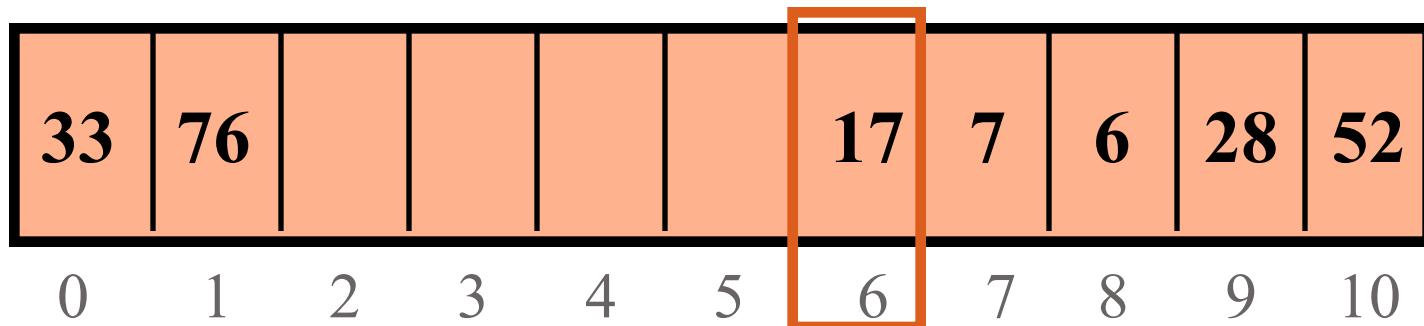
Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Suponer que ahora se da de baja al elemento **6**.



Ejemplo Prueba Lineal...

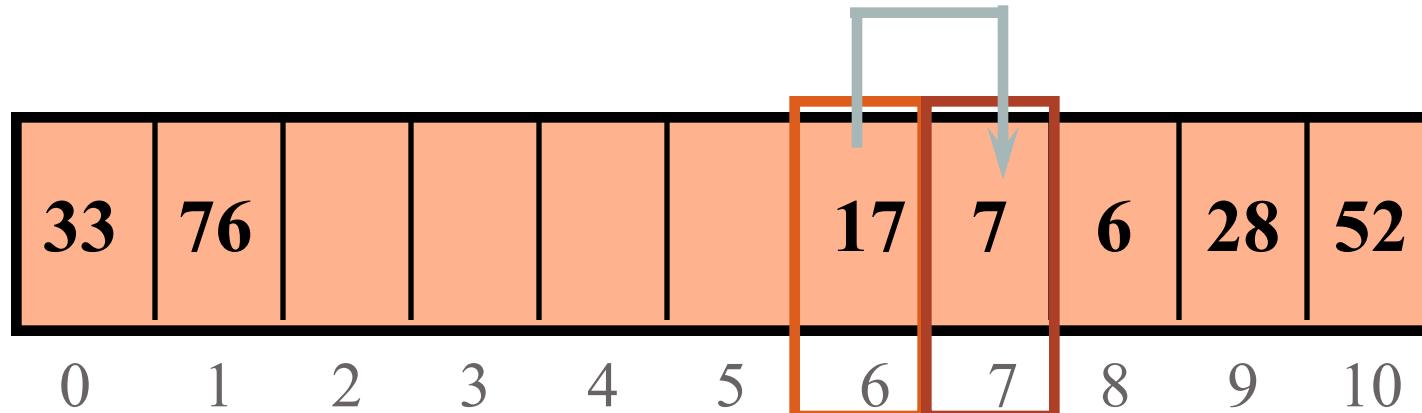
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Suponer que ahora se da de baja al elemento **6**.



NO ES EL ELEMENTO BUSCADO

Ejemplo Prueba Lineal...

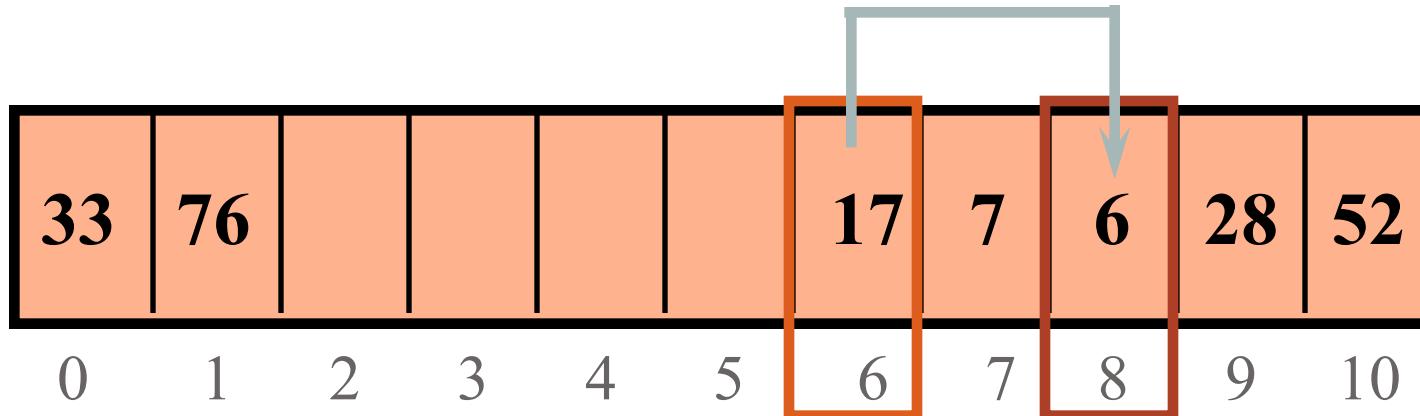
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Suponer que ahora se da de baja al elemento **6**.



NO ES EL ELEMENTO BUSCADO

Ejemplo Prueba Lineal...

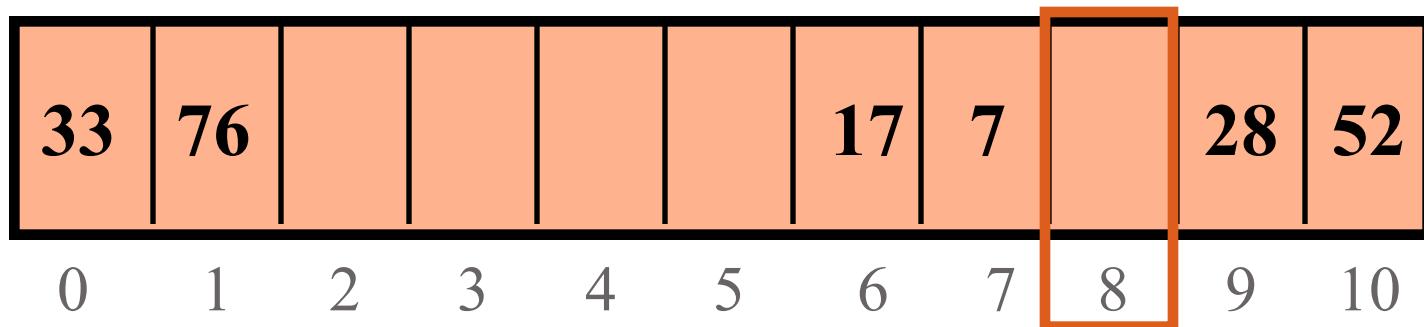
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Suponer que ahora se da de baja al elemento **6**.
 - Se busca el valor empleando la prueba lineal.
 - Se borra el elemento.



ELEMENTO ENCONTRADO

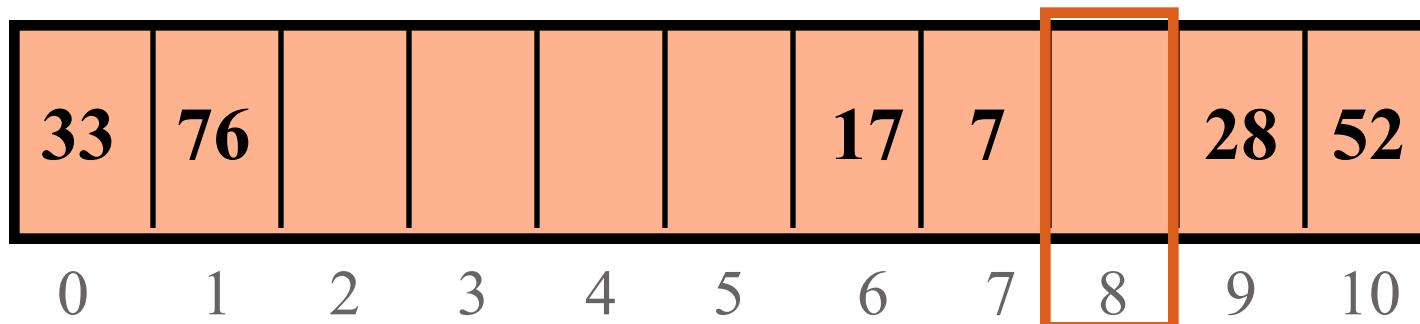
Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ Suponer que ahora se da de baja al elemento **6**.
 - Se busca el valor empleando la prueba lineal.
 - Se borra el elemento.



Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ ¿Qué pasaría después de borrar el **6** se quiere borrar el **52**?
 - **Problema:** Como se borró el 6 el espacio está vacío. Eso puede indicarnos que el 52 no está.
 - **Solución:** Una bandera de estado (status) que indique si la casilla está vacía, está ocupada o está borrada (porque ya alguna vez tuvo un dato).



PROBLEMAS

Prueba Lineal

► Estatus

- Cada casilla puede tener uno de los siguientes estatus:
 - Celda vacía
 - Celda borrada
 - Celda ocupada
- Uso del estatus cuando se busca, agrega o borra:
 - Para buscar un elemento, se compara casilla por casilla a partir de la casilla que generó la función de Hash. Se detiene al encontrar el elemento o hasta encontrar una casilla VACIA (no está).
 - Para agregar un elemento, se busca una casilla a partir de la casilla que generó la función de Hash. El elemento se almacena en una casilla VACIA o BORRADA.
 - Para borrar, se busca al elemento casilla por casilla a partir de la casilla que generó la función de Hash hasta encontrarlo o hasta llegar a una casilla VACIA (no está).

Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ ¿Qué pasaría ahora si se quisiera borrar al 52?

o = ocupado, v= vacío, b= borrado

o	o	v	v	v	v	o	o	b	o	o
33	76					17	7		28	52

0 1 2 3 4 5 6 7 8 9 10

CASILLA BORRADA, CONTINUAR

Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ ¿Qué pasaría ahora si se quisiera borrar al 52?

o = ocupado, v= vacío, b= borrado

o	o	v	v	v	v	o	o	b	o	o
33	76					17	7	28	52	

0 1 2 3 4 5 6 7 8 9 10

NO ES EL ELEMENTO BUSCADO

Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ ¿Qué pasaría ahora si se quisiera borrar al 52?

o = ocupado, v= vacío, b= borrado

o	o	v	v	v	v	o	o	b	o	o
33	76					17	7		28	52

0 1 2 3 4 5 6 7 8 9 10

VALOR ENCONTRADO

Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ ¿Qué pasaría ahora si se quisiera borrar al 52?

o = ocupado, v= vacío, b= borrado

o	o	v	v	v	v	o	o	b	o	b
33	76					17	7		28	

0 1 2 3 4 5 6 7 8 9 10

Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ ¿Qué pasa si se desea insertar el 39?

o = ocupado, v= vacío, b= borrado

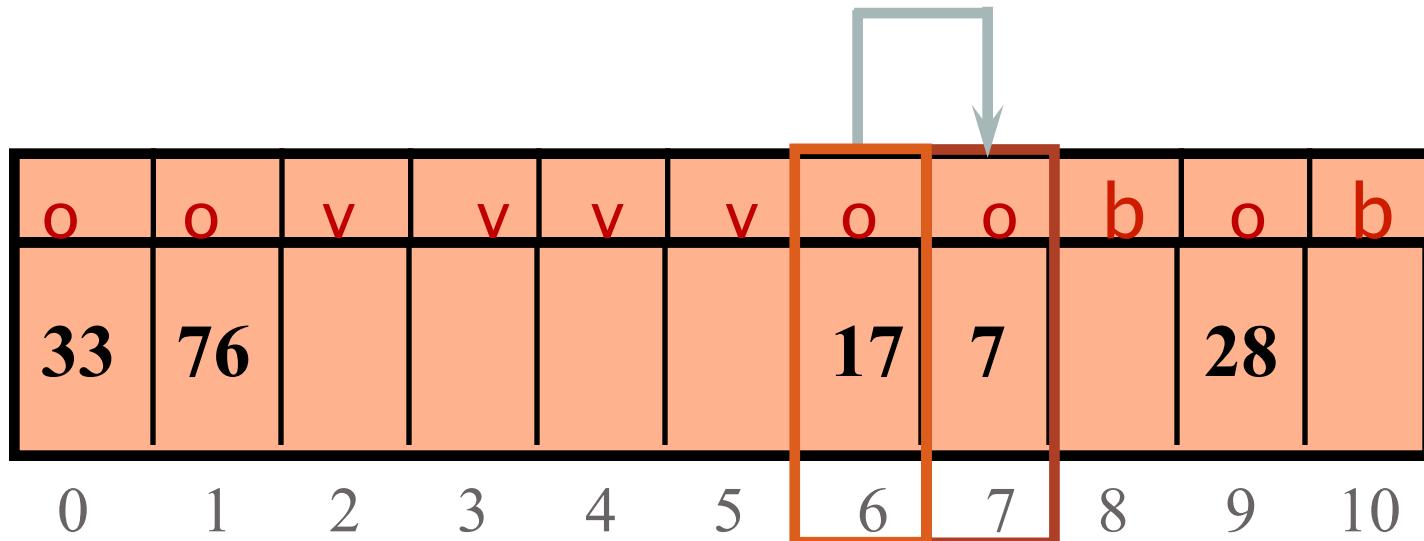
o	o	v	v	v	v	o	o	b	o	b
33	76					17	7		28	

0 1 2 3 4 5 6 7 8 9 10

COLISIÓN

Ejemplo Prueba Lineal...

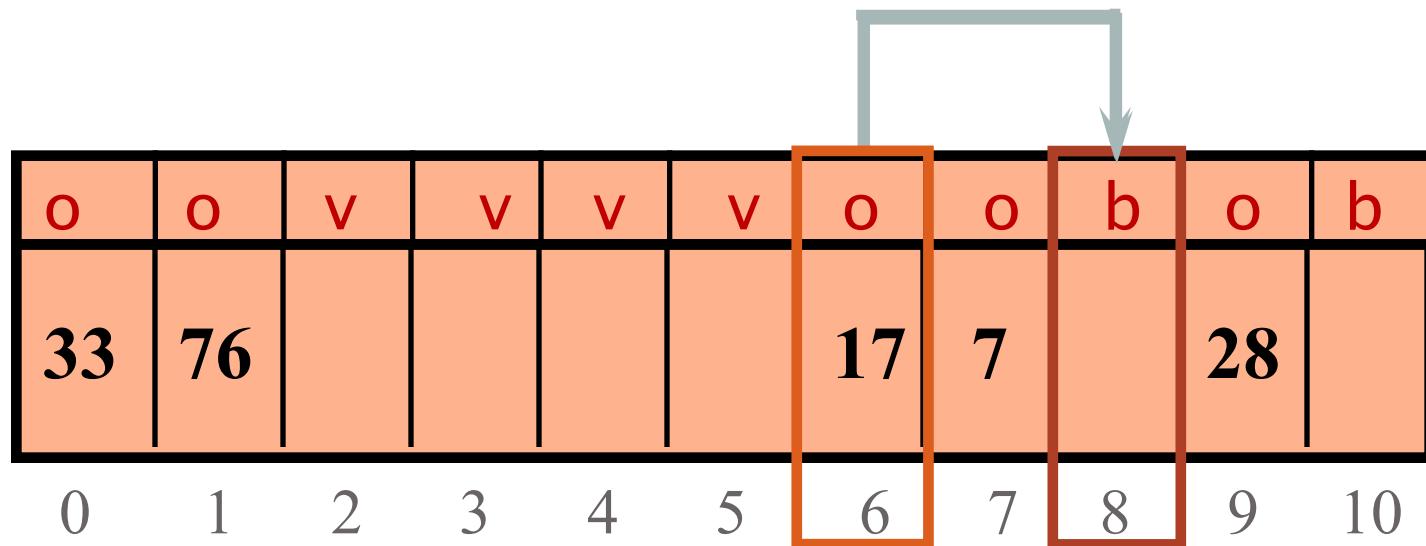
- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ ¿Qué pasa si se desea insertar el 39?
o = ocupado, v= vacío, b= borrado



COLISIÓN

Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ ¿Qué pasa si se desea insertar el 39?
o = ocupado, v= vacío, b= borrado



DISPONIBLE

Ejemplo Prueba Lineal...

- ▶ Suponer : $H(\text{llave}) = \text{llave \% } 11$
- ▶ ¿Qué pasa si se desea insertar el 39?
o = ocupado, v= vacío, b= borrado

o	o	v	v	v	v	o	o	o	o	b
33	76					17	7	39	28	

0 1 2 3 4 5 6 7 8 9 10

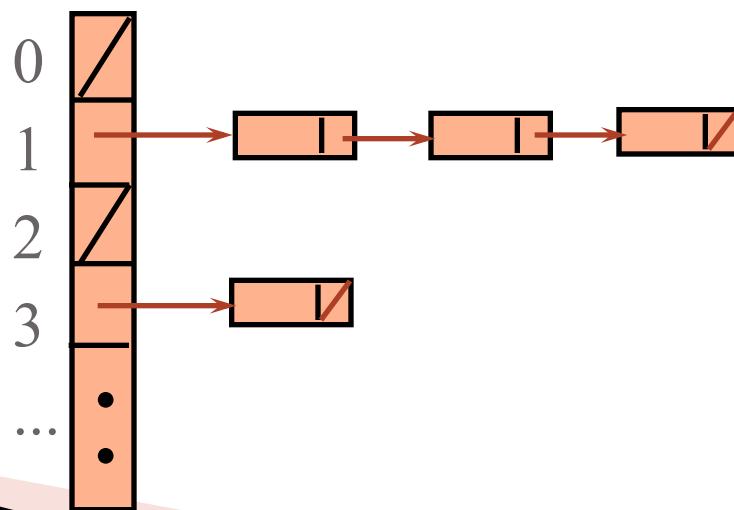
VALOR AGREGADO

Métodos de Encadenamiento.

- ▶ Los principales métodos son:
 - Método de Encadenamiento Externo.
 - Método de Encadenamiento de Colisiones.

Encadenamiento Externo.

- ▶ Cada casilla del arreglo es un *bucket* que apunta a una lista encadenada (puede ser un árbol u otra estructura de datos).
- ▶ La lista encadenada está formada por todos los elementos que colisionaron en la misma casilla.



Encadenamiento de Colisiones.

- ▶ Los elementos colisionados son acomodados en un espacio contiguo a la Tabla de Hash, denominado área de *Overlay*.

