Algoritmos y Estructuras de Datos I

Facultad de Ciencias Exactas y Tecnología
Universidad Nacional de Tucumán
2024

UNIVERSIDAD NACIONAL DE TUCUMÁN
facet
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍA

Búsqueda(3)

Dispersión (Hashing)

La primeras ideas del tema aparecieron por primera vez en 1953 mencionadas por H. P. Luhn en un memorándum de IBM, pero las primeras publicaciones recién aparecieron en 1956 en los libros de A. Dumey y en 1957 en los de W. Peterson.

Es interesante notar que la palabra “Hashing” nunca se usó, hasta que en 1968 fue publicada por R. Morris, a partir de entonces comenzó a ser una terminología estándar para la transformación de claves.

Significado según el diccionario Collins: *a technique for locating data in a file by applying a transformation, usually arithmetic, to a key.*

Traducción de Hash: **desplazar**, **trocear**, **picar**.

Dispersión (Hashing)

Otra manera de implementación de un ADT TABLA es la *dispersión* que permite hacer referencia de forma directa a la posición de un elemento dentro de la Tabla de registros en función de su clave.

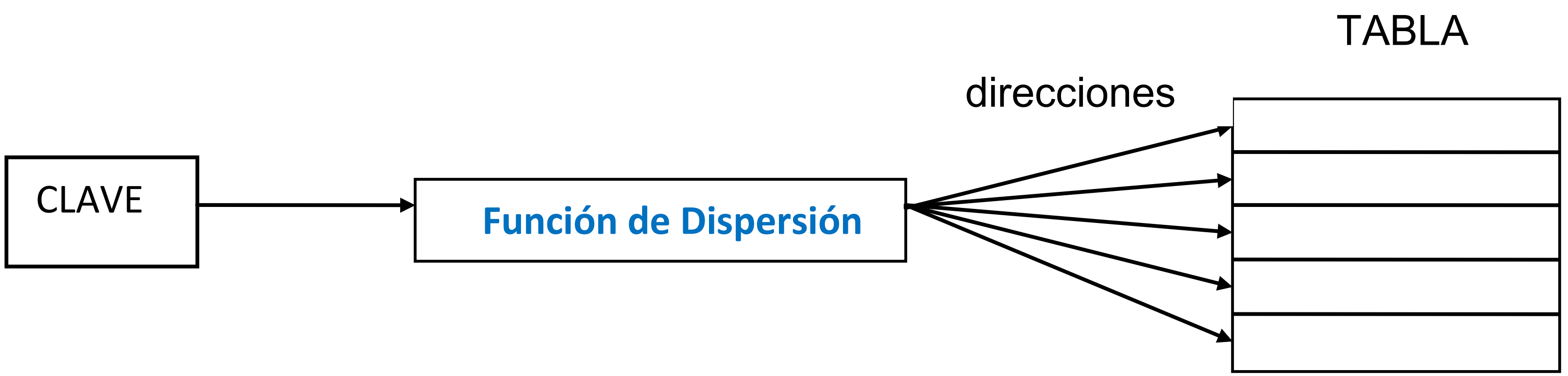
Este método permite hacer directamente referencia a los registros de la Tabla por medio de transformaciones aritméticas de las claves para obtener direcciones de la tabla.

Por ejemplo si se sabe que las claves son enteros distintos entre 1 y N, entonces se puede almacenar un registro de clave k en la posición k de la tabla, de modo que el acceso se hace en forma inmediata. La dispersión es una generalización de este método trivial en aplicaciones de búsqueda donde no se tiene ningún conocimiento concreto sobre el valor de las claves.

Dispersión (Hashing)

La implementación de un ADT TABLA con dispersión permite determinar de forma unívoca la posición de un elemento dentro de la Tabla en función de su clave.

El primer paso consiste en **usar una función de dispersión** que transforma la clave de búsqueda en direcciones de la Tabla.

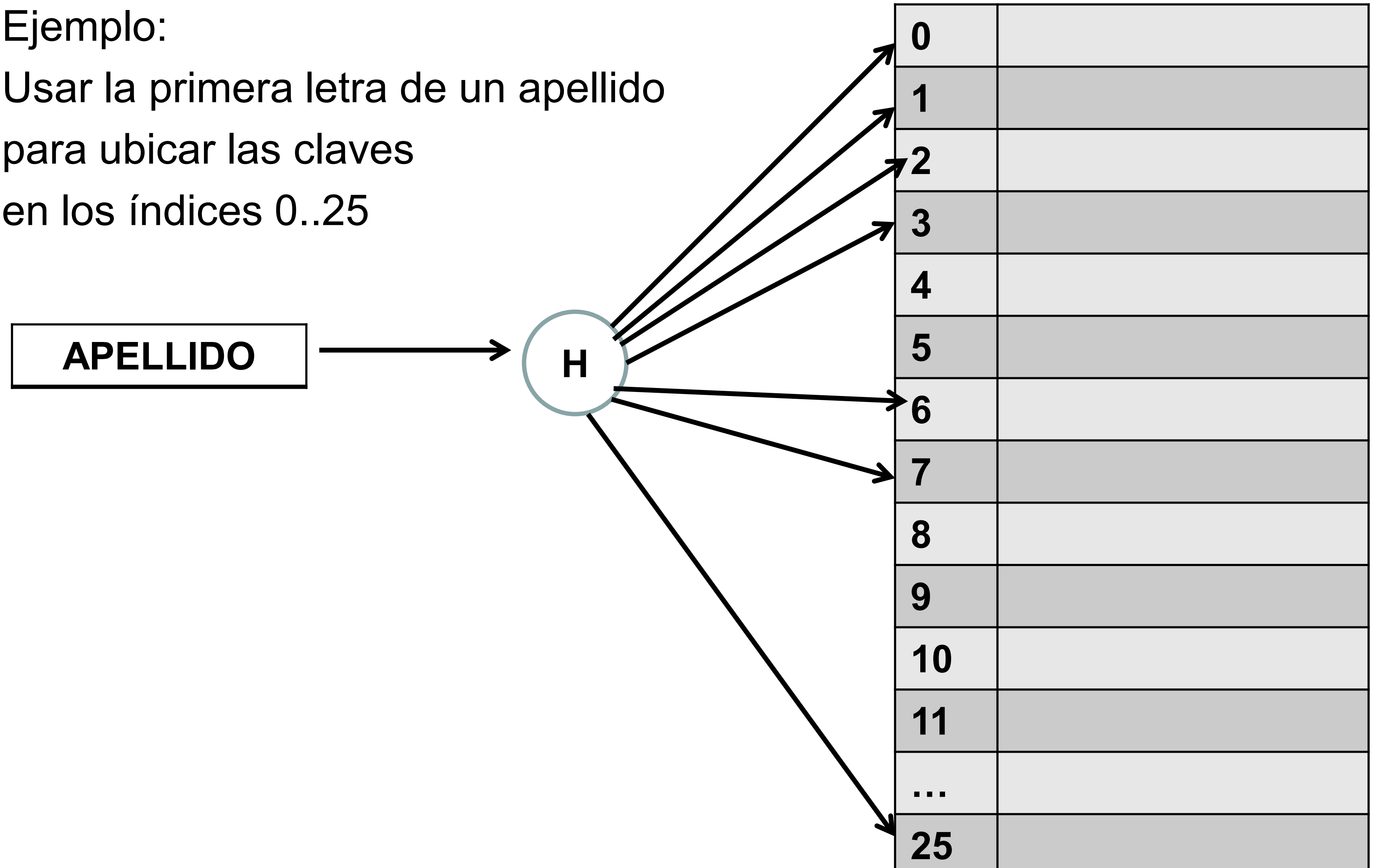


Al usar la función de dispersión dos o mas claves pueden dar la misma dirección en la tabla lo que se conoce como colisión.

El segundo paso es el proceso de **resolución de colisiones**

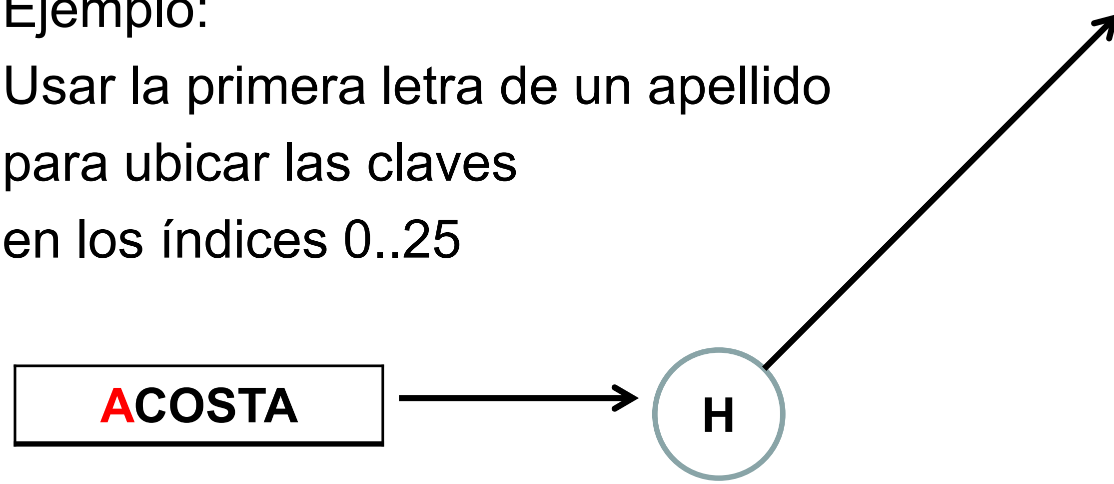
Dispersión (Hashing)

Ejemplo:
Usar la primera letra de un apellido
para ubicar las claves
en los índices 0..25



Dispersión (Hashing)

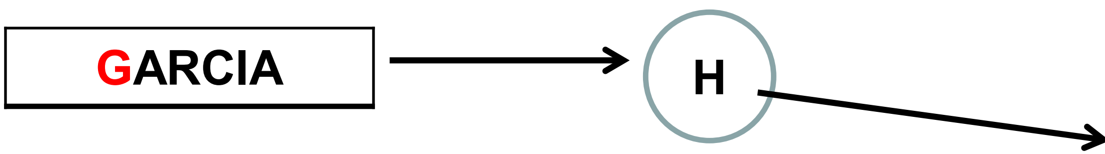
Ejemplo:
Usar la primera letra de un apellido
para ubicar las claves
en los índices 0..25



0	ACOSTA
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
…	
25	

Dispersión (Hashing)

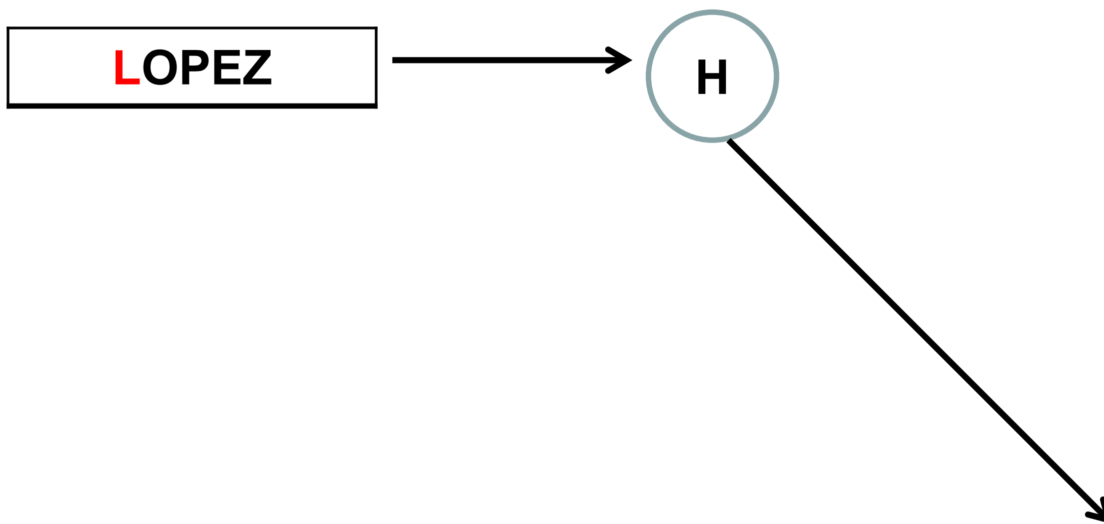
Ejemplo:
Usar la primera letra de un apellido
para ubicar las claves
en los índices 0..25



0	ACOSTA
1	
2	
3	
4	
5	
6	GARCIA
7	
8	
9	
10	
11	
…	
25	

Dispersión (Hashing)

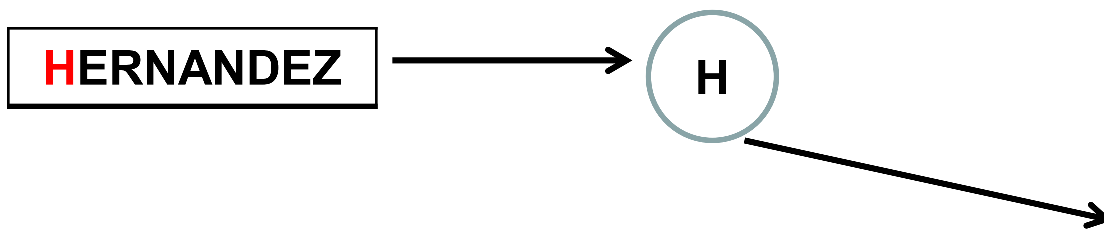
Ejemplo:
Usar la primera letra de un apellido
para ubicar las claves
en los índices 0..25



0	ACOSTA
1	
2	
3	
4	
5	
6	GARCIA
7	
8	
9	
10	
11	LOPEZ
…	
25	

Dispersión (Hashing)

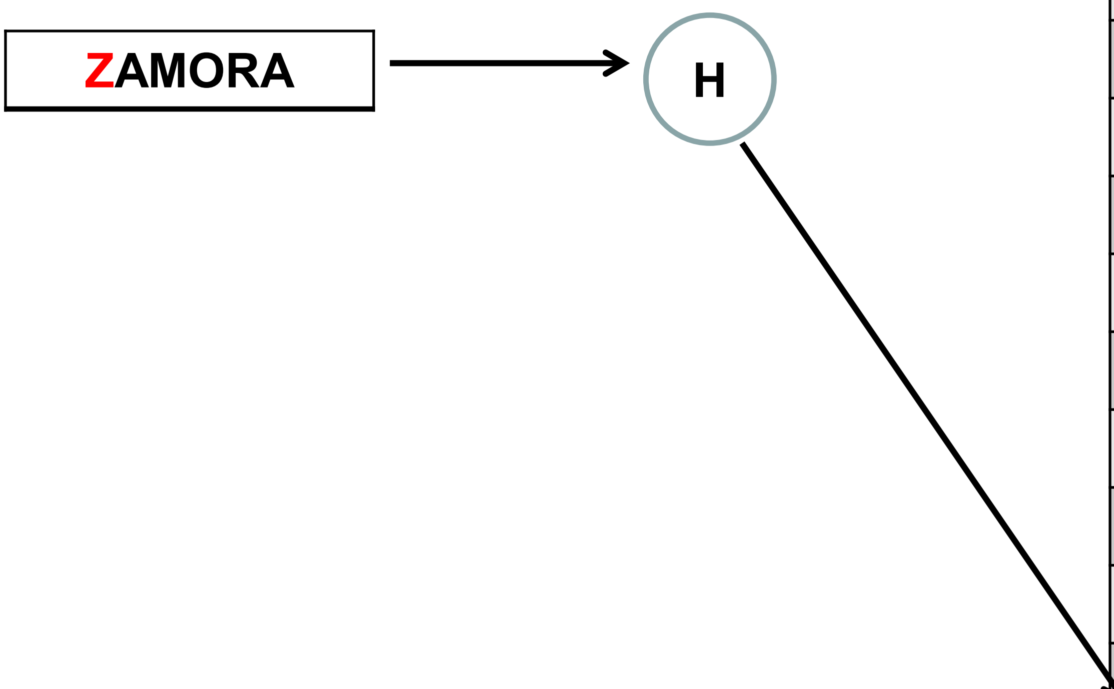
Ejemplo:
Usar la primera letra de un apellido
para ubicar las claves
en los índices 0..25



0	ACOSTA
1	
2	
3	
4	
5	
6	GARCIA
7	HERNANDEZ
8	
9	
10	
11	LOPEZ
…	
25	

Dispersión (Hashing)

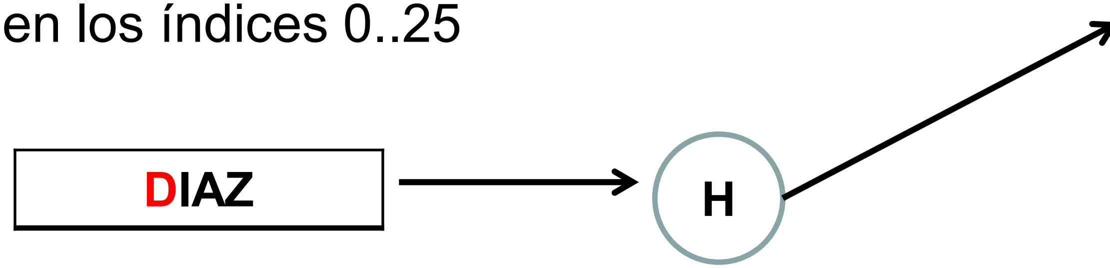
Ejemplo:
Usar la primera letra de un apellido
para ubicar las claves
en los índices 0..25



0	ACOSTA
1	
2	
3	
4	
5	
6	GARCIA
7	HERNANDEZ
8	
9	
10	
11	LOPEZ
…	
25	ZAMORA

Dispersión (Hashing)

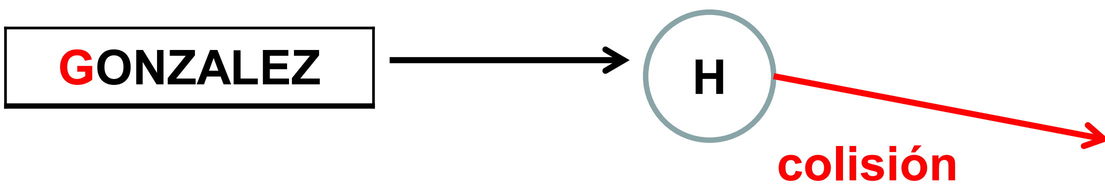
Ejemplo:
Usar la primera letra de un apellido
para ubicar las claves
en los índices 0..25



0	ACOSTA
1	
2	
3	DIAZ
4	
5	
6	GARCIA
7	HERNANDEZ
8	
9	
10	
11	LOPEZ
…	
25	ZAMORA

Dispersión (Hashing)

Ejemplo:
Usar la primera letra de un apellido para ubicar las claves en los índices 0..25



0	ACOSTA
1	
2	
3	DIAZ
4	
5	
6	GARCIA
7	HERNANDEZ
8	
9	
10	
11	LOPEZ
…	
25	ZAMORA

Donde se ubica GONZALEZ ?

Dispersión (Hashing)

Clásico compromiso espacio - tiempo

- Si no hay limitación de espacio, la función de dispersión es trivial la clave es el índice en una gran tabla, entonces se hace un solo acceso a la tabla.

- Si no hay limitación de tiempo, se puede resolver con un mínimo de memoria con un método de búsqueda secuencial.

- El mundo real: limitaciones de espacio y de tiempo.

- Objetivos básicos de la dispersión: uso eficaz de la memoria disponible y un acceso eficiente a la misma.

Funciones de Dispersión

- La función de dispersión permite calcular la dirección para comenzar una búsqueda.
- La función de dispersión h tiene dominio en K conjunto de claves y valores en el intervalo entero [0..M-1], siendo M el tamaño de la tabla:

$$\mathbf{h:} \qquad \mathbf{K \rightarrow \{0,1,2,\ldots,M-1\}}$$

de modo que dada una clave $k \in K$ entonces: $0 \leq h(k) < M$

- El éxito del método de dispersión reside en la elección de una buena función de transformación.

Características deseables de una función de dispersión:

- Debe ser una función ***fácil de calcular*****.**
- Debe ***distribuir las entradas en forma uniforme*** **en la tabla**.

Funciones de Dispersión
Dispersión por medio del cuadrado

La clave numérica se eleva al cuadrado, algunos dígitos específicos se extraen de la mitad del resultado para construir la dirección en la tabla.

Por ejemplo:

Valor de la Clave	*Clave al cuadrado*	*Dirección*
123456789	**1524157 8750 190521**	**8750**
987654321	**97546105 5789 971041**	**5789**
123456790	**1524157 8997 104100**	**8997**
555555555	**30864197 4691 358025**	**4691**
000000472	**0000000 0000 222784**	**0000** **** colisión
100064183	**1001286 0719 457489**	**0719**
200120472	**4004820 3313 502784**	**3313**
200120473	**4004820 3713 743729**	**3713**
117400000	**1378276 0000 000000**	**0000** **** colisión
027000400	**0243002 1600 160000**	**1600**

Funciones de Dispersión
Dispersión basada en la división

Es muy simple de calcular. Se usa el resto modulo M :

$$H(k)=k \bmod M$$

En este caso la elección de M es crítica, hay algunos valores mucho mejores que otros. En general, se sugiere que **M sea un número primo**, será la mejor elección en casi todos los casos.

Ejemplo 1:

Valor de la Clave	*Dirección = Clave mod 1000*
4967000	000
8421002	002
4618396	396
4587397	397
0004599	599
0000600	600
1234980	980
5555991	991
1234995	995
0001999	999

Funciones de Dispersión
Dispersión basada en la división

Ejemplo 2:

k	kmod100	kmod 97
212	12	18
618	18	36
302	2	11
940	40	67
702	2	23
704	4	25
612	12	30
606	6	24
772	72	93
510	10	25
423	23	35
650	50	68
317	17	26
907	7	34
507	7	22
304	4	13
714	14	35
857	57	81
801	1	25
900	0	27
413	13	25
701	1	22
418	18	30
601	1	19

Funciones de Dispersión
Dispersión por pliegue (folding)

Plegado con desplazamiento: se alinean los bits menos significativos de cada parte y se suma.

Ejemplo: Sea la clave k = 12320324111220, se la divide:

123	203	241	112	20

y luego se suma:

123
203
241
112
 20
----
699 → este será el valor de h(k)

Funciones de Dispersión
Dispersión por pliegue (folding)

Plegado al limite: el número se pliega en los limites de las partes, de modo que algunos dígitos que corresponden a posiciones consecutivas limites de partes se suman entre si.

Ejemplo: Sea la clave k = 12320324111220, se la divide:

123	203	241	112	20

Se acomoda y se suma:

123
302 → *orden inverso de 203*
241
211 → *orden inverso de 112*
 20
------
897 → este será el valor de h(k)

Dispersión
Factor de carga

Definición:

Se define el factor de carga de una tabla de dispersión como la fracción de la tabla que está ocupada.

Se calcula como el cociente entre el número de entradas ocupadas en la tabla(n) y el tamaño de la misma (M).

El factor de carga se denota con α

$$\alpha = n / M$$

Está demostrado que todas las funciones de dispersión trabajan pobremente cuando la tabla el factor de carga se aproxima a 1. Un factor de carga de 0.7 a 0.8 es casi lo máximo que puede tolerarse para un desempeño adecuado.

Dispersión y colisiones

Funciones que evitan duplicar valores son muy difíciles de conseguir aun que la tabla sea grande.

Por ejemplo la “paradoja de los cumpleaños” asevera que si 23 o más personas están en una habitación, hay buena chance de que dos de ellas tengan el mismo día y mes de cumpleaños.

*La **paradoja del cumpleaños** establece que si hay 23 personas reunidas hay una probablidad del 50,7% de que al menos dos personas de ellas cumplan años el mismo día. Para 60 o más personas la probabilidad es mayor del 99%.*

En este caso sería como transformar claves en una tabla de tamaño M=365, y en n=23 inserciones ya hay mas probabilidad de encontrar la primera colisión que de no encontrarla.

Dispersión y colisiones

Suponiendo una función aleatoria de hashing. ¿Cuántas inserciones se harán en la tabla hasta encontrar la primera colisión.?

M **(Tamaño de la tabla)**	**Número de inserciones hasta la 1ª. colisión**
100	12
1000	40
10000	125
M	$\sqrt{\pi \frac{M}{2}}$

Dispersión
Resolución de colisiones

- Las funciones de dispersión convierten las claves en direcciones de la tabla: **como resolver los casos en los que dos claves dan la misma dirección.?**
- Resolución de colisiones: desarrollar algoritmos y buscar estructuras de datos adecuadas para tratar dos claves que mapean en el mismo índice del arreglo.
- El gran desafío es resolver las colisiones en forma eficiente.

Hay dos tipos de soluciones:

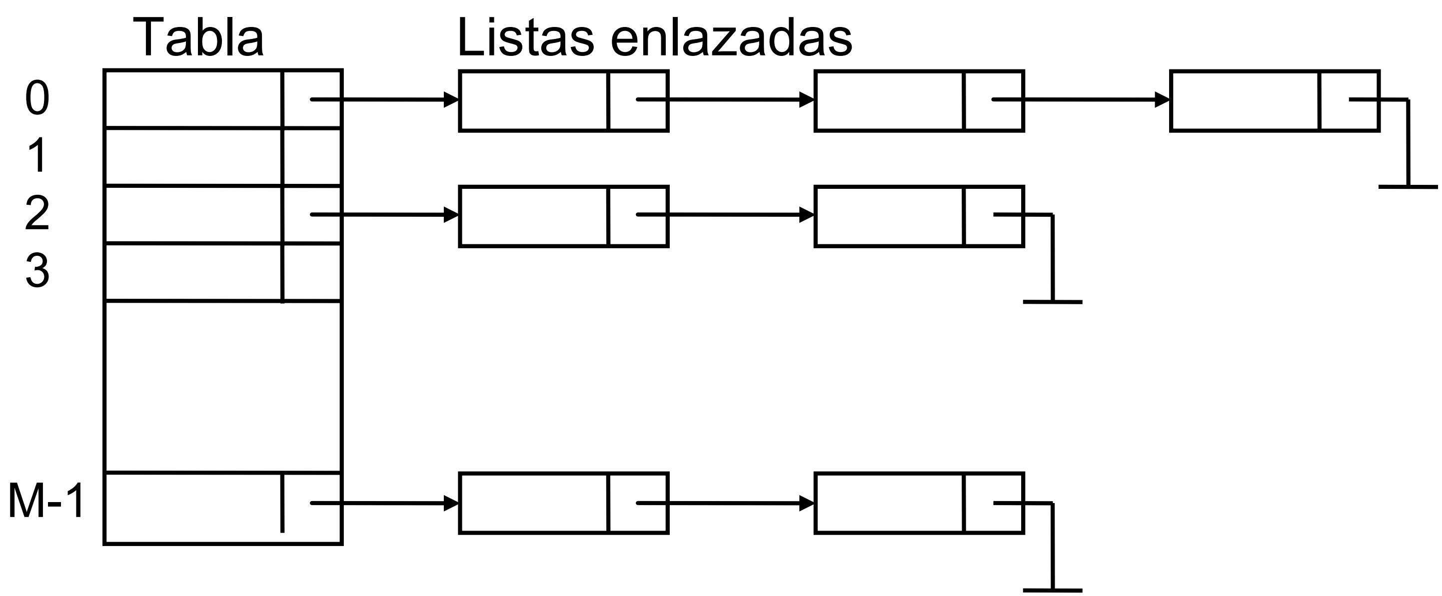
- **Dispersión abierta** que resuelve las colisiones con espacio de memoria adicional a la tabla
- **Dispersión cerrada** que resuelve el problema dentro de la misma tabla.

Dispersión
Resolución de colisiones

Dispersión abierta. Encadenamiento separado.

(H. P. Luhn, IBM 1953)

El método más directo para resolver colisiones consiste en construir para cada dirección de la tabla, una **lista enlazada con todos los registros cuyas claves se transforman esta dirección**.



Dispersión
Resolución de colisiones

Dispersión abierta.

- Usa una tabla de M<n listas enlazadas
- La función de dispersión mapea la clave a un índice i entre 0 y M-1

  H(clave)=i
- Insertar una clave: se ingresa la clave en un nodo al comienzo de la lista enlazada correspondiente al índice i de la tabla.
- Búsqueda de una clave: se busca solamente en la lista enlazada correspondiente al índice i

Dispersión
Resolución de colisiones

Ejemplo:

Datos: **SEARCHEXAMPLE**

Clave k	Información	Indice h(k)
S	0	2
E	1	0
A	2	0
R	3	4
C	4	4
H	5	4
E	6	0
X	7	2
A	8	0
M	9	4
P	10	3
L	11	3
E	12	0

TABLA DE DISPERSION

0	→
1	→
2	→ (S,0)
3	→
4	→

Dispersión

Resolución de colisiones

Ejemplo:

Datos: **SEARCHEXAMPLE**

Clave k	Información	Indice h(k)
S	0	2
E	1	0
A	2	0
R	3	4
C	4	4
H	5	4
E	6	0
X	7	2
A	8	0
M	9	4
P	10	3
L	11	3
E	12	0

TABLA DE DISPERSION

0	→ (E,1)
1	→
2	→ (S,0)
3	→
4	→

Dispersión
Resolución de colisiones

Ejemplo:

Datos: **SEARCHEXAMPLE**

Clave k	Información	Indice h(k)
S	0	**2**
E	1	**0**
A	2	**0**
R	3	**4**
C	4	**4**
H	5	**4**
E	6	**0**
X	7	**2**
A	8	**0**
M	9	**4**
P	10	**3**
L	11	**3**
E	12	**0**

TABLA DE DISPERSION

0	→ (A,2)→(E,1)
1	→
2	→ (S,0)
3	→
4	→

Dispersión
Resolución de colisiones

Ejemplo:

Datos: **SEARCHEXAMPLE**

Clave k	Información	Indice h(k)
S	0	**2**
E	1	**0**
A	2	**0**
R	3	**4**
C	4	**4**
H	5	**4**
E	6	**0**
X	7	**2**
A	8	**0**
M	9	**4**
P	10	**3**
L	11	**3**
E	12	**0**

TABLA DE DISPERSION

0	→ (A,2)→(E,1)
1	→
2	→ (S,0)
3	→
4	→ (R,3)

Dispersión
Resolución de colisiones

Ejemplo:

Datos: **SEARCHEXAMPLE**

Clave k	Información	Indice h(k)
S	0	**2**
E	1	**0**
A	2	**0**
R	3	**4**
C	4	**4**
H	5	**4**
E	6	**0**
X	7	**2**
A	8	**0**
M	9	**4**
P	10	**3**
L	11	**3**
E	12	**0**

TABLA DE DISPERSION

0	→ (A,2)→(E,1)
1	→
2	→ (S,0)
3	→
4	→ (C,4)→(R,3)

Dispersión
Resolución de colisiones

Ejemplo:

Datos: **SEARCHEXAMPLE**

Clave k	Información	Indice h(k)
S	0	**2**
E	1	**0**
A	2	**0**
R	3	**4**
C	4	**4**
H	5	**4**
E	6	**0**
X	7	**2**
A	8	**0**
M	9	**4**
P	10	**3**
L	11	**3**
E	12	**0**

TABLA DE DISPERSION

0	→(A,2)→(E,1)
1	→
2	→(S,0)
3	→
4	→(H,5)→(C,4)→(R,3)

Dispersión
Resolución de colisiones

Ejemplo:

Datos: **SEARCHEXAMPLE**

Clave k	Información	Indice h(k)
S	0	**2**
E	1	**0**
A	2	**0**
R	3	**4**
C	4	**4**
H	5	**4**
E	6	**0**
X	7	**2**
A	8	**0**
M	9	**4**
P	10	**3**
L	11	**3**
E	12	**0**

TABLA DE DISPERSION

0	→ (E,6)→(A,2)→(E,1)
1	→
2	→ (S,0)
3	→
4	→ (H,5)→(C,4)→(R,3)

Dispersión
Resolución de colisiones

Ejemplo:

Datos: **SEARCHEXAMPLE**

Clave k	Información	Indice h(k)
S	0	**2**
E	1	**0**
A	2	**0**
R	3	**4**
C	4	**4**
H	5	**4**
E	6	**0**
X	7	**2**
A	8	**0**
M	9	**4**
P	10	**3**
L	11	**3**
E	12	**0**

TABLA DE DISPERSION

0	→ (E,6)→(A,2)→(E,1)
1	→
2	→ (X,7)→(S,0)
3	→
4	→(H,5)→(C,4)→(R,3)

Dispersión
Resolución de colisiones

Ejemplo:

Datos: **SEARCHEXAMPLE**

Clave k	Información	Indice h(k)
S	0	**2**
E	1	**0**
A	2	**0**
R	3	**4**
C	4	**4**
H	5	**4**
E	6	**0**
X	7	**2**
A	8	**0**
M	9	**4**
P	10	**3**
L	11	**3**
E	12	**0**

TABLA DE DISPERSION

0	→ (A,8)→(E,6)→(A,2)→(E,1)
1	→
2	→ (X,7)→(S,0)
3	→
4	→(H,5)→(C,4)→(R,3)

Dispersión
Resolución de colisiones

Ejemplo:

Datos: **SEARCHEXAMPLE**

Clave k	Información	Indice h(k)
S	0	**2**
E	1	**0**
A	2	**0**
R	3	**4**
C	4	**4**
H	5	**4**
E	6	**0**
X	7	**2**
A	8	**0**
M	9	**4**
P	10	**3**
L	11	**3**
E	12	**0**

TABLA DE DISPERSION

0	→ (A,8) → (E,6) → (A,2) → (E,1)
1	→
2	→ (X,7) → (S,0)
3	→
4	→ (M,9) → (H,5) → (C,4) → (R,3)

Dispersión
Resolución de colisiones

Ejemplo:

Datos: **SEARCHEXAMPLE**

Clave k	Información	Indice h(k)
S	0	2
E	1	0
A	2	0
R	3	4
C	4	4
H	5	4
E	6	0
X	7	2
A	8	0
M	9	4
P	10	3
L	11	3
E	12	0

TABLA DE DISPERSION

0	→(A,8)→(E,6)→(A,2)→(E,1)
1	→
2	→ (X,7)→(S,0)
3	→(P,10)
4	→ (M,9)→(H,5)→(C,4)→(R,3)

Dispersión
Resolución de colisiones

Ejemplo:

Datos: **SEARCHEXAMPLE**

Clave k	Información	Indice h(k)
S	0	**2**
E	1	**0**
A	2	**0**
R	3	**4**
C	4	**4**
H	5	**4**
E	6	**0**
X	7	**2**
A	8	**0**
M	9	**4**
P	10	**3**
L	11	**3**
E	12	**0**

TABLA DE DISPERSION

0	→(A,8)→(E,6)→(A,2)→(E,1)
1	→
2	→ (X,7)→(S,0)
3	→ (L,11)→(P,10)
4	→ (M,9)→(H,5)→(C,4)→(R,3)

Dispersión
Resolución de colisiones

Ejemplo:

Datos: **SEARCHEXAMPLE**

Clave k	Información	Indice h(k)
S	0	**2**
E	1	**0**
A	2	**0**
R	3	**4**
C	4	**4**
H	5	**4**
E	6	**0**
X	7	**2**
A	8	**0**
M	9	**4**
P	10	**3**
L	11	**3**
E	12	**0**

TABLA DE DISPERSION

0	→(E,12)→(A,8)→(E,6)→(A,2)→(E,1)
1	→
2	→ (X,7)→(S,0)
3	→ (L,11)→(P,10)
4	→ (M,9)→(H,5)→(C,4)→(R,3)

Dispersión
Resolución de colisiones

Ejemplo:

Datos: **SEARCHEXAMPLE**

Clave k	Información	Indice h(k)
S	0	2
E	1	0
A	2	0
R	3	4
C	4	4
H	5	4
E	6	0
X	7	2
A	8	0
M	9	4
P	10	3
L	11	3
E	12	0

TABLA DE DISPERSION

0	→ (E,12) → (A,8) → (E,6) → (A,2) → (E,1)
1	→
2	→ (X,7) → (S,0)
3	→ (L,11) → (P,10)
4	→ (M,9) → (H,5) → (C,4) → (R,3)

Dispersión
Resolución de colisiones

Dispersión cerrada. Prueba Lineal.
[Amdahl-Boehme-Rocherster-Samuel, IBM 1953]

El método más simple de direccionamiento que se puede implementar es la llamada **exploración lineal** , o sondeo lineal o prueba lineal, (linear probing)

Cuando hay una colisión, en la operación de INSERCIÓN, se explora la siguientes posiciones de la tabla, hasta encontrar el primer lugar libre y allí se ubica la clave.
Así la secuencia de exploración es :

< h(k), h(k)+1, … M-1, 0, 1,…, h(k)-2, h(k)-1>

Dispersión
Resolución de colisiones

Prueba Lineal:

- El tamaño de la tabla M debe ser mayor que n.
- La función de dispersión mapea la clave a un índice i dentro de la tabla, entre 0 y M-1:

H(clave)=i

- Inserción de una clave : se ingresa la clave en el índice i de la tabla si esta libre, sino se intenta con i+1, i+2 etc.
- Búsqueda de una clave: se busca en el índice i de la tabla, si está ocupado pero no coincide se intenta con i+1, i+2, etc.. Cuidado con las posiciones libres.
- Desventaja: los ítems tienden a agruparse en distintas sectores de la tabla. Este efecto se conoce como agrupamiento primario (primary clustering).

Dispersión
Resolución de colisiones

Prueba Lineal:

Bajo la suposición de hashing uniforme, la fórmula para el número promedio de exploraciones necesarias, expresada en función del factor de carga α, para prueba lineal es:

$B1 = ½ (1+1/(1-\alpha)^2)$ para inserciones y búsquedas sin éxito.
$B2 = ½ (1+1/(1-\alpha))$ para búsquedas con éxito.

Si :	$\alpha = 1/2$	**B1 = 2.5**	**B2 = 1.5**
	$\alpha = 2/3$	**B1 = 5**	**B2 = 2**
	$\alpha = 3/4$	**B1 = 8.5**	**B2 = 2.5**
	$\alpha = 9/10$	**B1 = 50.5**	**B2 = 5**

Esto muestra que a medida que la tabla se va llenando (α se aproxima a 1), los valores de B se van haciendo más grandes.

Dispersión
Resolución de colisiones

Dispersión doble:

En este caso se utiliza una segunda función de dispersión para obtener un incremento fijo a usar en la secuencia de exploración.

Se aplica la función de dispersión primaria H1, si resulta una colisión, se usa la segunda H2 sucesivamente hasta encontrar una posición vacía. La segunda función de dispersión se debe elegir con cuidado.

Por ejemplo
si: **H1(k) = k mod M**,
se puede elegir: **H2(k) = R - (k mod R)** donde R es un número primo menor que M.

Otra buena elección sería: **H2(k) = 1 + k/M mod (M-2)**

Dispersión
Resolución de colisiones

Dispersión doble:

La fórmula exacta para el número medio de exploraciones que se hacen en la técnica de doble dispersión con una función de doble dispersión "independiente" es:

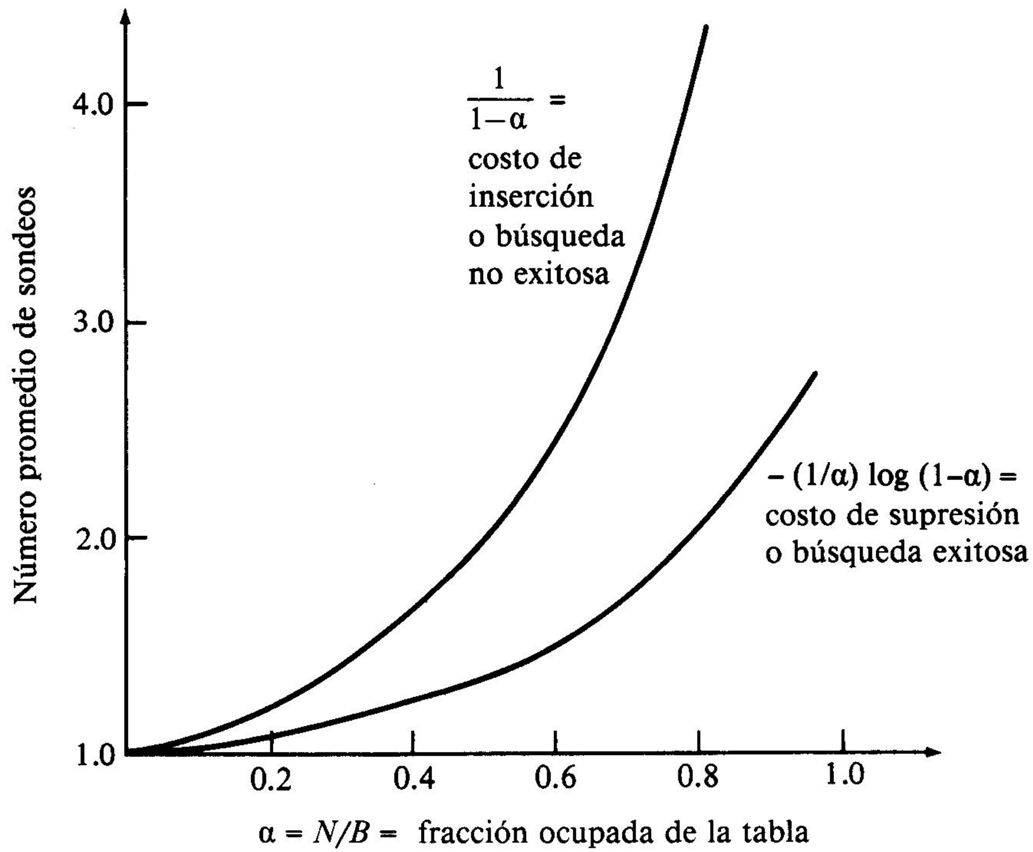
$B1 = 1 / (1 - \alpha)$ para un búsqueda sin éxito
$B2 = - (\ln (1 - \alpha)) / \alpha$ para un búsqueda con éxito

Si:			
	$\alpha = 1/2$	**B1 = 2**	**B2 = 1.4**
	$\alpha = 2/3$	**B1 = 3**	**B2 = 1.6**
	$\alpha = 3/4$	**B1 = 4**	**B2 = 1.8**
	$\alpha = 9/10$	**B1 = 10**	**B2 = 2.5**

En la práctica, esto significa que con la doble dispersión se puede utilizar una tabla más pequeña para lograr los mismos tiempos de búsqueda que con la exploración lineal.

Dispersión
Resolución de colisiones

Dispersión doble:



Dispersión
Dispersión Perfecta

Dado un conjunto de claves $k_1, k_2, \ldots, k_n$, una función de ***dispersión perfecta*** es una función tal que:

$$\mathbf{H(k_i) \neq H(k_j) \quad \forall\ i \neq j}$$

Cuando una función de *dispersión es perfecta* **no ocurren colisiones.**

Es difícil encontrar una función de dispersión perfecta para un conjunto particular de claves. Mientras más grande sea la tabla respecto al número de claves a insertar, más fácil será encontrar una función de dispersión perfecta.

En general es deseable tener una función de dispersión perfecta para un conjunto de n claves en una tabla de n posiciones. Una función de dispersión perfecta como esa se llama **mínima**. En la práctica esto es difícil de obtener.-