Juan Menduiña

Trabajo Práctico N° 5:
Hashing.

PARTE I: Preguntas Conceptuales.

Ejercicio 1.

Definir el concepto de hashing (o dispersión). ¿Cómo se relaciona este concepto con archivos?

Hashing (o dispersión) es una técnica utilizada para transformar una clave (como una palabra, un número o un conjunto de datos) en una dirección o índice dentro de una estructura de datos, generalmente una tabla o un archivo. Esta transformación se realiza mediante una función de dispersión (función *hash*), que toma la clave como entrada y produce un número entero, la dirección *hash* o el código *hash*, que se utiliza como índice para acceder, rápidamente, a los datos.

Hashing (o dispersión) es un método de asignación de claves a posiciones en una tabla o archivo mediante una función de dispersión (función *hash*), con el objetivo de facilitar el almacenamiento y recuperación eficiente de información.

- Técnica para generar una direción base única para una llave dada. La dispersión se usa cuando se requiere acceso rápido a una llave.
- Técnica que convierte la llave del registro en un número aleatorio, el que sirve, después, para determinar dónde se almacena el registro.
- Técnica de almacenamiento y recuperación que usa una función de *hash* para mapear registros en dirección de almacenamiento.

Ejercicio 2.

Explicar el concepto de función de dispersión. Enumerar, al menos, tres funciones de dispersión y explicar, brevemente, cómo funciona cada una.

Una función de dispersión (función *hash*) es un procedimiento que toma una clave (como una palabra, un número o un conjunto de datos) como entrada y la convierte en un número entero, la dirección *hash* o el código *hash*, que se utiliza como índice en una tabla o en un archivo para acceder, rápidamente, a los datos.

El objetivo principal es asignar claves a posiciones de manera uniforme para facilitar el acceso rápido a los datos, minimizando las colisiones (casos en los que diferentes claves generan el mismo índice).

Tres funciones de dispersión son:

- Método de la división: Toma la clave *k* (por ejemplo, un número entero) y la divide por un número *m* (preferentemente primo), y se usa el resto como la dirección *hash*.
  hash(k)= k mod m.
- Método de la multiplicación: Toma la clave *k* (por ejemplo, un númeor entero), la multiplica por un número fraccionario (A= 0,618), toma la parte decimal, y la multiplica por un número *m* para obtener un índice.
  hash(k)= floor(m * ((k * A) mod 1)).
- Método de la suma de caracteres (para cadenas): Suma los valores ASCII (o numéricos) de todos los caracteres de la clave y, luego, aplica una operación como *mod m*.

Ejercicio 3.

Explicar los conceptos de sinónimo, colisión y desborde (overflow). ¿Qué condición es necesaria en el archivo directo para que pueda ocurrir una colisión y no un desborde?

Sinónimo: Es una clave diferente que, al ser procesada por la función de dispersión, produce la misma dirección *hash* que otra clave. Es decir, dos claves distintas generan el mismo índice.

Colisión: Situación en la que un registro es asignado a una dirección que está utilizada por otro registro. Se produce cuando dos o más claves se asignan a la misma posición en la tabla o en el archivo.

Desborde (*overflow*): Situación en la que un registro es asignado a una dirección que está utilizada por otro registro y no queda espacio para este nuevo. Se produce cuando no hay espacio disponible en la posición calculada por la función de dispersión ni en las posiciones alternativas previstas para manejar colisiones.

La condición que es necesaria en el archivo directo para que pueda ocurrir una colisión y no un desborde es que exista, al menos, una posición disponible donde se pueda reubicar el sinónimo.

Ejercicio 4.

¿Qué alternativas existen para reducir el número de colisiones (y, por ende, de desbordes) en un archivo organizado mediante la técnica de hashing?

Las alternativas que existen para reducir el número de colisiones (y, por ende, de desbordes) en un archivo organizado mediante la técnica de *hashing* son:

- Algoritmos de dispersión sin colisiones o que éstas nunca produzcan *overflow* (perfectos e imposibles de conseguir).
- Buscar métodos que distribuyan los registros de la forma más aleatoria posible.
- Distribuir pocos registros en muchas direcciones.
- Colocar más de un registro por dirección.

Ejercicio 5.

Explicar, brevemente, qué es la densidad de empaquetamiento. ¿Cuáles son las consecuencias de tener una menor densidad de empaquetamiento en un archivo directo?

La densidad de empaquetamiento es la proporción de espacio del archivo asignado que, en realidad, almacena registros. Las consecuencias de tener una menor densidad de empaquetamiento en un archivo directo es menos *overflow* y más desperdicio de espacio.

Ejercicio 6.

Explicar, brevemente, cómo funcionan las siguientes técnicas de resolución de desbordes que se pueden utilizar en hashing estático: saturación progresiva, saturación encadenada, saturación progresiva encadenada con área de desborde separada y dispersión doble.

Saturación progresiva: Cuando se completa el nodo, se busca el próximo hasta encontrar uno libre.

Saturación encadenada: Es similar a la saturación progresiva, pero los registros de saturación se encadenan y no ocupan, necesariamente, posiciones contiguas.

Saturación progresiva encadenada con área de desborde separada: No utiliza nodos de direcciones para los *overflows*, éstos van a nodos especiales.

Dispersión doble: Las técnicas de saturación tienden a agrupar en zonas contiguas y generan búsquedas largas cuando la densidad tiende a uno. La solución de esta técnica de resolución de colisiones es almacenar los registros de *overflow* en zonas no relacionadas, aplicándoles una segunda función *hash* a la llave para producir un número entero, el cual se suma a la dirección original tantas veces como sea necesario hasta encontrar una dirección con espacio.

PARTE II: Dispersión Extensible.

Ejercicio 7.

Para las siguientes claves, realizar el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada nodo tiene capacidad para dos registros. El número natural indica el orden de llegada de las operaciones. Se debe mostrar el estado del archivo para cada operación. Justificar, brevemente, ante colisión y desborde los pasos que se realizan.

1	*+ Darín*	*00111111*	*2*	*+ Alterio*	*11110100*
3	*+ Sbaraglia*	*10100101*	*4*	*+ De la Serna*	*01010111*
5	*+ Altavista*	*01101011*	*6*	*+ Grandinetti*	*10101010*
7	*- Altavista*	*01101011*	*8*	*- Sbaraglia*	*10100101*

Estado inicial de la tabla de dispersión y del archivo:

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0		

Inserción de la clave “Darín”:

- No se produce colisión ni *overflow* en el bloque 0.

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0	Darín (00111111)	

Inserción de la clave “Alterio”:

- Se produce colisión en el bloque 0, pero no se produce *overflow*.

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0	Darín (00111111)	Alterio (11110100)

Inserción de la clave “Sbaraglia”:

- Se produce colisión y *overflow* en el bloque 0.
- Se genera el bloque 1.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 1	
Sufijo	**#Bloque**
(0)	1
(1)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1	Darín (00111111)	Sbaraglia (10100101)
1	1	Alterio (11110100)	

Inserción de la clave “De la Serna”:

- Se produce colisión y *overflow* en el bloque 0.
- Se genera el bloque 2.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	1
(01)	2
(10)	1
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Darín (00111111)	De la Serna (01010111)
1	1	Alterio (11110100)	
2	2	Sbaraglia (10100101)	

Inserción de la clave “Altavista”:

- Se produce colisión y *overflow* en el bloque 0.
- Se genera el bloque 3.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 3	
Sufijo	**#Bloque**
(000)	1
(001)	2
(010)	1
(011)	3
(100)	1
(101)	2
(110)	1
(111)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	3	Darín (00111111)	De la Serna (01010111)
1	1	Alterio (11110100)	
2	2	Sbaraglia (10100101)	
3	3	Altavista (01101011)	

Inserción de la clave “Grandinetti”:

- Se produce colisión en el bloque 1, pero no se produce *overflow*.

Bits de dispersión: 3	
Sufijo	**#Bloque**
(000)	1
(001)	2
(010)	1
(011)	3
(100)	1
(101)	2
(110)	1
(111)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	3	Darín (00111111)	De la Serna (01010111)
1	1	Alterio (11110100)	Grandinetti (10101010)
2	2	Sbaraglia (10100101)	
3	3	Altavista (01101011)	

Baja de la clave “Altavista”:

- Se borra la clave “Altavista”.
- Se libera el bloque 3, ya que se puede fusionar con el bloque 0.
- Se reducen en uno los bits del bloque 0.
- Se reduce a la mitad la cantidad de celdas de la tabla de dispersión y disminuyen en uno los bits de dispersión.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	1
(01)	2
(10)	1
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Darín (00111111)	De la Serna (01010111)
1	1	Alterio (11110100)	Grandinetti (10101010)
2	2	Sbaraglia (10100101)	
3	~~3~~	~~Altavista (01101011)~~	

Baja de la clave “Sbaraglia”:

- Se borra la clave “Sbaraglia”.
- Se libera el bloque 2, ya que se puede fusionar con el bloque 0.
- Se reducen en uno los bits del bloque 0.
- Se reduce a la mitad la cantidad de celdas de la tabla de dispersión y disminuyen en uno los bits de dispersión.

Bits de dispersión: 1	
Sufijo	**#Bloque**
(0)	1
(1)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1	Darín (00111111)	De la Serna (01010111)
1	1	Alterio (11110100)	Grandinetti (10101010)
2	~~2~~	~~Sbaraglia (10100101)~~	
3	~~3~~	~~Altavista (01101011)~~	

Ejercicio 8.

Realizar el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada nodo tiene capacidad para dos claves. El número natural indica el orden de llegada de las operaciones. Se debe mostrar el estado del archivo para cada operación. Justificar, brevemente, ante colisión y desborde los pasos que se realizan.

1	*+ Buenos Aires*	*...1001*	*2*	*+ San Juan*	*...0100*
3	*+ Entre Ríos*	*...1110*	*4*	*+ Corrientes*	*...0010*
5	*+ San Luis*	*...0101*	*6*	*+ Tucumán*	*...0111*
7	*+ Río Negro*	*...0011*	*8*	*+ Jujuy*	*...1111*
9	*+ Salta*	*...1010*	*10*	*- Río Negro*	*...0011*

Estado inicial de la tabla de dispersión y del archivo:

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0		

Inserción de la clave “Buenos Aires”:

- No se produce colisión ni *overflow* en el bloque 0.

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0	Buenos Aires (…1001)	

Inserción de la clave “San Juan”:

- Se produce colisión en el bloque 0, pero no se produce *overflow*.

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0	Buenos Aires (…1001)	San Juan (…0100)

Inserción de la clave “Entre Ríos”:

- Se produce colisión y *overflow* en el bloque 0.
- Se genera el bloque 1.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 1	
Sufijo	**#Bloque**
(0)	1
(1)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1	Buenos Aires (…1001)	
1	1	San Juan (…0100)	Entre Ríos (…1110)

Inserción de la clave “Corrientes”:

- Se produce colisión y *overflow* en el bloque 1.
- Se genera el bloque 2.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	2
(01)	0
(10)	1
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1	Buenos Aires (…1001)	
1	2	Entre Ríos (…1110)	Corrientes (…0010)
2	2	San Juan (…0100)	

Inserción de la clave “San Luis”:

- Se produce colisión en el bloque 0, pero no se produce *overflow*.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	2
(01)	0
(10)	1
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1	Buenos Aires (…1001)	San Luis (…0101)
1	2	Entre Ríos (…1110)	Corrientes (…0010)
2	2	San Juan (…0100)	

Inserción de la clave “Tucumán”:

- Se produce colisión y *overflow* en el bloque 0.
- Se genera el bloque 3.
- No se duplica la cantidad de celdas de la tabla de dispersión y no aumentan en uno los bits de dispersión.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	2
(01)	3
(10)	1
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Tucumán (…0111)	
1	2	Entre Ríos (…1110)	Corrientes (…0010)
2	2	San Juan (…0100)	
3	2	Buenos Aires (…1001)	San Luis (…0101)

Inserción de la clave “Río Negro”:

- Se produce colisión en el bloque 0, pero no se produce *overflow*.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	2
(01)	3
(10)	1
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Tucumán (…0111)	Río Negro (…0011)
1	2	Entre Ríos (…1110)	Corrientes (…0010)
2	2	San Juan (…0100)	
3	2	Buenos Aires (…1001)	San Luis (…0101)

Inserción de la clave “Jujuy”:

- Se produce colisión y *overflow* en el bloque 0.
- Se genera el bloque 4.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 3	
Sufijo	**#Bloque**
(000)	2
(001)	3
(010)	1
(011)	4
(100)	2
(101)	3
(110)	1
(111)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	3	Tucumán (…0111)	Jujuy (…1111)
1	2	Entre Ríos (…1110)	Corrientes (…0010)
2	2	San Juan (…0100)	
3	2	Buenos Aires (…1001)	San Luis (…0101)
4	3	Río Negro (…0011)	

Inserción de la clave “Salta”:

- Se produce colisión y *overflow* en el bloque 1.
- Se genera el bloque 5.
- No se duplica la cantidad de celdas de la tabla de dispersión y no aumentan en uno los bits de dispersión.

Bits de dispersión: 3	
Sufijo	**#Bloque**
(000)	2
(001)	3
(010)	5
(011)	4
(100)	2
(101)	3
(110)	1
(111)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	3	Tucumán (…0111)	Jujuy (…1111)
1	3	Entre Ríos (…1110)	
2	2	San Juan (…0100)	
3	2	Buenos Aires (…1001)	San Luis (…0101)
4	3	Río Negro (…0011)	
5	3	Corrientes (…0010)	Salta (…1010)

Baja de la clave “Río Negro”:

- Se borra la clave “Río Negro”.
- Se libera el bloque 4, ya que se puede fusionar con el bloque 0.
- Se reducen en uno los bits del bloque 0.

Bits de dispersión: 3	
Sufijo	**#Bloque**
(000)	2
(001)	3
(010)	5
(011)	0
(100)	2
(101)	3
(110)	1
(111)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Tucumán (…0111)	Jujuy (…1111)
1	3	Entre Ríos (…1110)	
2	2	San Juan (…0100)	
3	2	Buenos Aires (…1001)	San Luis (…0101)
4	~~3~~	~~Río Negro (…0011)~~	
5	3	Corrientes (…0010)	Salta (…1010)

Ejercicio 9.

Para las siguientes claves, realizar el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada nodo tiene capacidad para dos registros. El número natural indica el orden de llegada de las operaciones. Se debe mostrar el estado del archivo para cada operación. Justificar, brevemente, ante colisión y desborde los pasos que se realizan.

1	*+ Tristana*	*11110010*	*2*	*+ Jarvan IV*	*00111010*
3	*+ Teemo*	*01010100*	*4*	*+ Annie*	*10100101*
5	*+ Ryze*	*10101110*	*6*	*+ Morgana*	*01101011*
7	*+ Garen*	*11001011*	*8*	*- Teemo*	*01010100*

Estado inicial de la tabla de dispersión y del archivo:

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0		

Inserción de la clave “Tristana”:

- No se produce colisión ni *overflow* en el bloque 0.

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0	Tristana (11110010)	

Inserción de la clave “Jarvan IV”:

- Se produce colisión en el bloque, pero no se produce *overflow*.

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0	Tristana (11110010)	Jarvan IV (00111010)

Inserción de la clave “Teemo”:

- Se produce colisión y *overflow* en el bloque 0.
- Se genera el bloque 1.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.
- No son suficientes los bits de dispersión para redistribuir las claves.
- Se produce colisión y *overflow* en el bloque 1.
- Se genera el bloque 2.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	2
(01)	0
(10)	1
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1		
1	2	Tristana (11110010)	Jarvan IV (00111010)
2	2	Teemo (01010100)	

Inserción de la clave “Annie”:

- No se produce colisión ni *overflow* en el bloque 0.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	2
(01)	0
(10)	1
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1	Annie (10100101)	
1	2	Tristana (11110010)	Jarvan IV (00111010)
2	2	Teemo (01010100)	

Inserción de la clave “Ryze”:

- Se produce colisión y *overflow* en el bloque 1.
- Se genera el bloque 3.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 3	
Sufijo	**#Bloque**
(000)	2
(001)	0
(010)	3
(011)	0
(100)	2
(101)	0
(110)	1
(111)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1	Annie (10100101)	
1	3	Ryze (10101110)	
2	2	Teemo (01010100)	
3	3	Tristana (11110010)	Jarvan IV (00111010)

Inserción de la clave “Morgana”:

- Se produce colisión en el bloque 0, pero no se produce *overflow*.

Bits de dispersión: 3	
Sufijo	**#Bloque**
(000)	2
(001)	0
(010)	3
(011)	0
(100)	2
(101)	0
(110)	1
(111)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1	Annie (10100101)	Morgana (01101011)
1	3	Ryze (10101110)	
2	2	Teemo (01010100)	
3	3	Tristana (11110010)	Jarvan IV (00111010)

Inserción de la clave “Garen”:

- Se produce colisión y *overflow* en el bloque 0.
- Se genera el bloque 4.
- No se duplica la cantidad de celdas de la tabla de dispersión y no aumentan los bits de dispersión.

Bits de dispersión: 3	
Sufijo	**#Bloque**
(000)	2
(001)	4
(010)	3
(011)	0
(100)	2
(101)	4
(110)	1
(111)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Morgana (01101011)	Garen (11001011)
1	3	Ryze (10101110)	
2	2	Teemo (01010100)	
3	3	Tristana (11110010)	Jarvan IV (00111010)
4	2	Annie (10100101)	

Baja de la clave “Teemo”:

- Se borra la clave “Teemo”.
- No se libera el bloque 2, ya que no se puede fusionar.

Bits de dispersión: 3	
Sufijo	**#Bloque**
(000)	2
(001)	4
(010)	3
(011)	0
(100)	2
(101)	4
(110)	1
(111)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Morgana (01101011)	Garen (11001011)
1	3	Ryze (10101110)	
2	2	~~Teemo (01010100)~~	
3	3	Tristana (11110010)	Jarvan IV (00111010)
4	2	Annie (10100101)	

Ejercicio 10.

Para las siguientes claves, realizar el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada nodo tiene capacidad para dos registros. El número natural indica el orden de llegada de las operaciones. Se debe mostrar el estado del archivo para cada operación. Justificar, brevemente, ante colisión y desborde los pasos que se realizan.

1	*+ Guillermo.B*	*01100011*	*2*	*+ Gómez*	*00000001*
3	*+ Gustavo.B*	*01010110*	*4*	*+ Sosa*	*11110100*
5	*+ Enría*	*00110101*	*6*	*+ Guli*	*00101000*
7	*- Gustavo.B*	*01010110*	*8*	*- Sosa*	*11110100*

Estado inicial de la tabla de dispersión y del archivo:

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0		

Inserción de la clave “Guillermo.B”:

- No se produce colisión ni *overflow* en el bloque 0.

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0	Guillermo.B (01100011)	

Inserción de la clave “Gómez”:

- Se produce colisión en el bloque 0, pero no se produce *overflow*.

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0	Guillermo.B (01100011)	Gómez (00000001)

Inserción de la clave “Gustavo.B”:

- Se produce colisión y *overflow* en el bloque 0.
- Se genera el bloque 1.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 1	
Sufijo	**#Bloque**
(0)	1
(1)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1	Guillermo.B (01100011)	Gómez (00000001)
1	1	Gustavo.B (01010110)	

Inserción de la clave “Sosa”:

- Se produce colisión en el bloque 1, pero no se produce *overflow*.

Bits de dispersión: 1	
Sufijo	**#Bloque**
(0)	1
(1)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1	Guillermo.B (01100011)	Gómez (00000001)
1	1	Gustavo.B (01010110)	Sosa (11110100)

Inserción de la clave “Enria”:

- Se produce colisión y *overflow* en el bloque 0.
- Se genera el bloque 2.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	1
(01)	2
(10)	1
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Guillermo.B (01100011)	
1	1	Gustavo.B (01010110)	Sosa (11110100)
2	2	Gómez (00000001)	Enria (00110101)

Inserción de la clave “Guli”:

- Se produce colisión y *overflow* en el bloque 1.
- Se genera el bloque 3.
- No se duplica la cantidad de celdas de la tabla de dispersión y no aumentan en uno los bits de dispersión.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	3
(01)	2
(10)	1
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Guillermo.B (01100011)	
1	2	Gustavo.B (01010110)	
2	2	Gómez (00000001)	Enria (00110101)
3	2	Sosa (11110100)	Guli (00101000)

Baja de la clave “Gustavo.B”:

- Se borra la clave “Gustavo.B”.
- Se libera el bloque 1, ya que se puede fusionar con el bloque 3.
- Se reducen en uno los bits del bloque 3.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	3
(01)	2
(10)	3
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Guillermo.B (01100011)	
1	~~2~~	~~Gustavo.B (01010110)~~	
2	2	Gómez (00000001)	Enria (00110101)
3	1	Sosa (11110100)	Guli (00101000)

Baja de la clave “Sosa”:

- Se borra la clave “Sosa”.
- No se libera el bloque 3, ya que sigue ocupado por la clave “Guli”.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	3
(01)	2
(10)	3
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Guillermo.B (01100011)	
1	~~2~~	~~Gustavo.B (01010110)~~	
2	2	Gómez (00000001)	Enria (00110101)
3	1	~~Sosa (11110100)~~	Guli (00101000)

Ejercicio 11.

Para las siguientes claves, realizar el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada nodo tiene capacidad para dos registros. El número natural indica el orden de llegada de las operaciones. Se debe mostrar el estado del archivo para cada operación. Justificar, brevemente, ante colisión y desborde los pasos que se realizan.

1	*+ Mansilla*	*01100010*	*2*	*+ Cetré*	*10001000*
3	*+ Ascacibar*	*01010111*	*4*	*+ Carrillo*	*11110101*
5	*+ Manyoma*	*00110100*	*6*	*+ Méndez*	*00101001*
7	*+ Alario*	*11000101*	*8*	*- Mansilla*	*01100010*

Estado inicial de la tabla de dispersión y del archivo:

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0		

Inserción de la clave “Mansilla”:

- No se produce colisión ni *overflow* en el bloque 0.

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0	Mansilla (01100010)	

Inserción de la clave “Cetré”:

- Se produce colisión en el bloque 0, pero no se produce *overflow*.

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0	Mansilla (01100010)	Cetré (10001000)

Inserción de la clave “Ascacibar”:

- Se produce colisión y *overflow* en el bloque 0.
- Se genera el bloque 1.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 1	
Sufijo	**#Bloque**
(0)	1
(1)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1	Ascacibar (01010111)	
1	1	Mansilla (01100010)	Cetré (10001000)

Inserción de la clave “Carrillo”:

- Se produce colisión en el bloque 0, pero no se produce *overflow*.

Bits de dispersión: 1	
Sufijo	**#Bloque**
(0)	1
(1)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1	Ascacibar (01010111)	Carrillo (11110101)
1	1	Mansilla (01100010)	Cetré (10001000)

Inserción de la clave “Manyoma”:

- Se produce colisión y *overflow* en el bloque 1.
- Se genera el bloque 2.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	2
(01)	0
(10)	1
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1	Ascacibar (01010111)	Carrillo (11110101)
1	2	Mansilla (01100010)	
2	2	Cetré (10001000)	Manyoma (00110100)

Inserción de la clave “Méndez”:

- Se produce colisión y *overflow* en el bloque 0.
- Se genera el bloque 3.
- No se duplica la cantidad de celdas de la tabla de dispersión y no aumentan en uno los bits de dispersión.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	2
(01)	3
(10)	1
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Ascacibar (01010111)	
1	2	Mansilla (01100010)	
2	2	Cetré (10001000)	Manyoma (00110100)
3	2	Carrillo (11110101)	Méndez (00101001)

Inserción de la clave “Alario”:

- Se produce colisión y *overflow* en el bloque 3.
- Se genera el bloque 4.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 3	
Sufijo	**#Bloque**
(000)	2
(001)	4
(010)	1
(011)	0
(100)	2
(101)	3
(110)	1
(111)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Ascacibar (01010111)	
1	2	Mansilla (01100010)	
2	2	Cetré (10001000)	Manyoma (00110100)
3	3	Carrillo (11110101)	Alario (11000101)
4	3	Méndez (00101001)	

Baja de la clave “Mansilla”:

- Se borra la clave “Mansilla”.
- Se libera el bloque 1, ya que se puede fusionar con el bloque 2.
- Se reducen en uno los bits del bloque 2.

Bits de dispersión: 3	
Sufijo	**#Bloque**
(000)	2
(001)	4
(010)	2
(011)	0
(100)	2
(101)	3
(110)	2
(111)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Ascacibar (01010111)	
1	~~2~~	~~Mansilla (01100010)~~	
2	1	Cetré (10001000)	Manyoma (00110100)
3	3	Carrillo (11110101)	Alario (11000101)
4	3	Méndez (00101001)	

Ejercicio 12.

Realizar el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada nodo tiene capacidad para dos claves. El número natural indica el orden de llegada de las operaciones. Se deberán explicar los pasos que se realizan en cada operación y dibujar los estados sucesivos correspondientes (inclusive el estado inicial).

1	*+ Aconcagua*	*10100111*	*2*	*+ Kilimanjaro*	*10101010*
3	*+ Mont Blanc*	*00111110*	*4*	*+ Cervino*	*01101111*
5	*+ Etna*	*00110101*	*6*	*+ Chañi*	*11110000*
7	*+ Cho Oyu*	*01011101*	*8*	*+ Vinicunca*	*01011011*
9	*- Chañi*	*11110000*	*10*	*- Cervino*	*01101111*

Estado inicial de la tabla de dispersión y del archivo:

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0		

Inserción de la clave “Aconcagua”:

- No se produce colisión ni *overflow* en el bloque 0.

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0	Aconcagua (10100111)	

Inserción de la clave “Kilimanjaro”:

- Se produce colisión en el bloque 0, pero no se produce *overflow*.

Bits de dispersión: 0	
Sufijo	**#Bloque**
(0)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	0	Aconcagua (10100111)	Kilimanjaro (10101010)

Inserción de la clave “Mont Blanc”:

- Se produce colisión y *overflow* en el bloque 0.
- Se genera el bloque 1.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 1	
Sufijo	**#Bloque**
(0)	1
(1)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1	Aconcagua (10100111)	
1	1	Kilimanjaro (10101010)	Mont Blanc (00111110)

Inserción de la clave “Cervino”:

- Se produce colisión en el bloque 0, pero no se produce *overflow*.

Bits de dispersión: 1	
Sufijo	**#Bloque**
(0)	1
(1)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	1	Aconcagua (10100111)	Cervino (01101111)
1	1	Kilimanjaro (10101010)	Mont Blanc (00111110)

Inserción de la clave “Etna”:

- Se produce colisión y *overflow* en el bloque 0.
- Se genera el bloque 2.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	1
(01)	2
(10)	1
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Aconcagua (10100111)	Cervino (01101111)
1	1	Kilimanjaro (10101010)	Mont Blanc (00111110)
2	2	Etna (00110101)	

Inserción de la clave “Chañi”:

- Se produce colisión y *overflow* en el bloque 1.
- Se genera el bloque 3.
- No se duplica la cantidad de celdas de la tabla de dispersión y no aumentan en uno los bits de dispersión.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	3
(01)	2
(10)	1
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Aconcagua (10100111)	Cervino (01101111)
1	2	Kilimanjaro (10101010)	Mont Blanc (00111110)
2	2	Etna (00110101)	
3	2	Chañi (11110000)	

Inserción de la clave “Cho Oyu”:

- Se produce colisión en el bloque 2, pero no se produce *overflow*.

Bits de dispersión: 2	
Sufijo	**#Bloque**
(00)	3
(01)	2
(10)	1
(11)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	2	Aconcagua (10100111)	Cervino (01101111)
1	2	Kilimanjaro (10101010)	Mont Blanc (00111110)
2	2	Etna (00110101)	Cho Oyu (01011101)
3	2	Chañi (11110000)	

Inserción de la clave “Vinicunca”:

- Se produce colisión y *overflow* en el bloque 0.
- Se genera el bloque 4.
- Se duplica la cantidad de celdas de la tabla de dispersión y aumentan en uno los bits de dispersión.

Bits de dispersión: 3	
Sufijo	**#Bloque**
(000)	3
(001)	2
(010)	1
(011)	4
(100)	3
(101)	2
(110)	1
(111)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	3	Aconcagua (10100111)	Cervino (01101111)
1	2	Kilimanjaro (10101010)	Mont Blanc (00111110)
2	2	Etna (00110101)	Cho Oyu (01011101)
3	2	Chañi (11110000)	
4	3	Vinicunca (01011011)	

Baja de la clave “Chañi”:

- Se borra la clave “Chañi”.
- Se libera el bloque 3, ya que se puede fusionar con el bloque 1.
- Se reducen en uno los bits del bloque 1.

Bits de dispersión: 3	
Sufijo	**#Bloque**
(000)	1
(001)	2
(010)	1
(011)	4
(100)	1
(101)	2
(110)	1
(111)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	3	Aconcagua (10100111)	Cervino (01101111)
1	1	Kilimanjaro (10101010)	Mont Blanc (00111110)
2	2	Etna (00110101)	Cho Oyu (01011101)
3	~~2~~	~~Chañi (11110000)~~	
4	3	Vinicunca (01011011)	

Baja de la clave “Cervino”:

- Se borra la clave “Cervino”.
- No se libera el bloque 0, ya que sigue ocupado por la clave “Aconcagua”.

Bits de dispersión: 3	
Sufijo	**#Bloque**
(000)	1
(001)	2
(010)	1
(011)	4
(100)	1
(101)	2
(110)	1
(111)	0

#Bloque	**Bits**	**Clave R1**	**Clave R2**
0	3	Aconcagua (10100111)	~~Cervino (01101111)~~
1	1	Kilimanjaro (10101010)	Mont Blanc (00111110)
2	2	Etna (00110101)	Cho Oyu (01011101)
3	~~2~~	~~Chañi (11110000)~~	
4	3	Vinicunca (01011011)	