

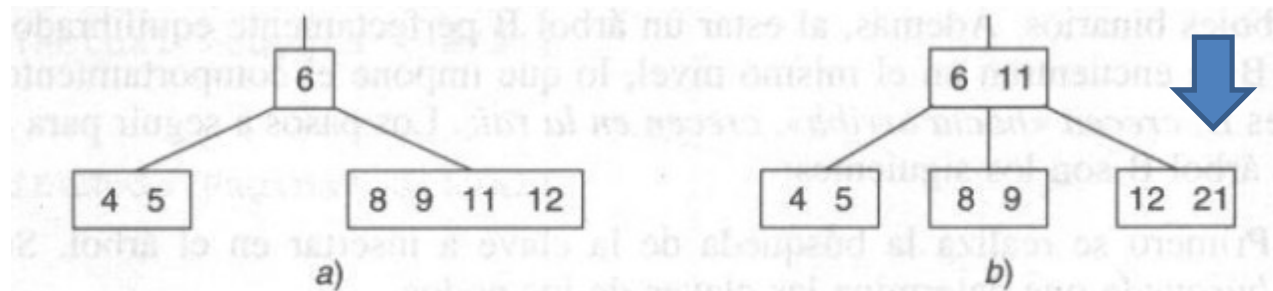
Árboles B

Capítulo 16 , Joyanes

16.3. PROCESO DE FORMACIÓN DE UN ÁRBOL B

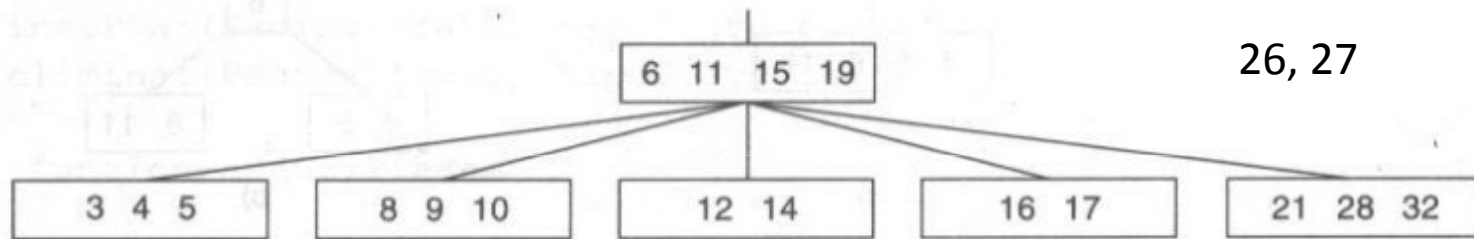
Las claves que se añaden a un árbol B siempre se insertan a partir de un nodo hoja, como ocurre en los árboles binarios. Además, al estar un árbol B perfectamente equilibrado, todas las hojas de un árbol B se encuentran en el mismo nivel, lo que impone el comportamiento característico de los árboles B: *crecen «hacia arriba», crecen en la raíz*. Los pasos a seguir para añadir una nueva clave en un árbol B son los siguientes:

- Primero se realiza la búsqueda de la clave a insertar en el árbol. Se sigue el *camino de búsqueda* que determina las claves de los nodos.
- En el caso de que la clave no esté en el árbol, la búsqueda termina en un nodo *hoja*. Entonces, empieza el proceso de inserción de la nueva clave.
- De no estar lleno el nodo hoja, la inserción es posible en dicho nodo y termina la inserción.
- En caso de estar llena la *hoja*, la inserción no es posible en dicho nodo; el comportamiento característico de los árboles B se pone de manifiesto, entonces se divide el nodo en dos en el mismo nivel del árbol, excepto la clave *mediana*, que no se incluye en ninguno de los dos nodos, sino que *sube* en el árbol por el *camino de búsqueda*, para a su vez repetir el proceso de inserción en el nodo *antecedente*. Por esto un **árbol B crece hacia arriba**; esta *ascensión* de la clave *mediana* puede ocurrir que llegue al nodo raíz, entonces éste se divide en dos nodos y la clave enviada hacia arriba se convierte en una nueva raíz. Ésta es la forma en que el árbol B crece en altura.



21

Figura 16.3. Inserción de nuevas claves en un árbol B de orden 5.



26, 27

Figura 16.5. Árbol B de orden 5 después de una división de nodo.

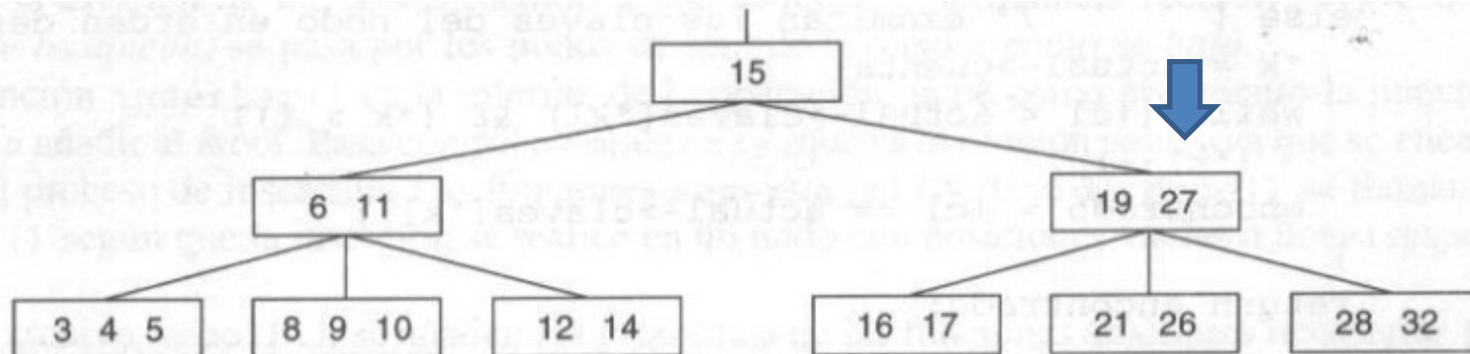


Figura 16.6. Árbol B de orden 5 con todas las claves.

16.6. ELIMINACIÓN DE UNA CLAVE EN UN ÁRBOL B

Ésta es otra de las operaciones básicas del tipo abstracto árbol B. El objetivo de la operación es que dada una clave de un árbol, extraerla del nodo en que se encuentra con la condición de que la estructura resultante siga siendo árbol B. De igual forma que la inserción de una clave siempre se hace en una página *hoja*, la eliminación también se hace en una página (nodo) *hoja*, debido a que de esa manera se mantiene más fácilmente la condición de que las claves de un nodo dividan a las de sus descendientes a la manera de un árbol de búsqueda.

Es evidente que dada una clave cualquiera no tiene por qué estar ubicada en un nodo *hoja*, sino que puede estar en cualquier nodo interno. Entonces se aplica la siguiente propiedad de los árboles B: *si una clave no está en una hoja, su clave predecesora o sucesora en el árbol B (predecesora o sucesora en el orden natural) se encuentra en un nodo hoja*. Esta propiedad se puede observar en el árbol B de la Figura 16.8; por ejemplo, si se elige la clave del nodo raíz, 45, la clave sucesora en el árbol es 48, que como se puede observar está ubicada en un nodo *hoja*; lo mismo se puede decir de la clave predecesora, 32, ubicada en otro nodo *hoja*.

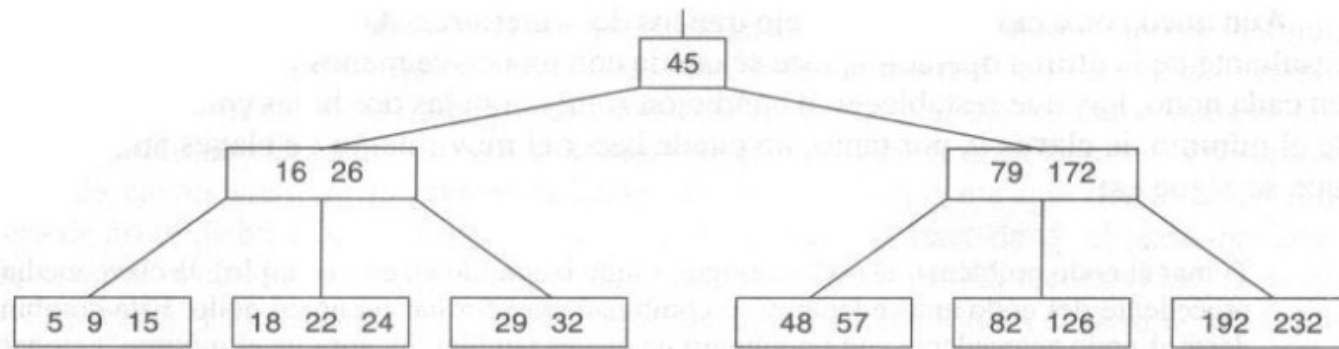
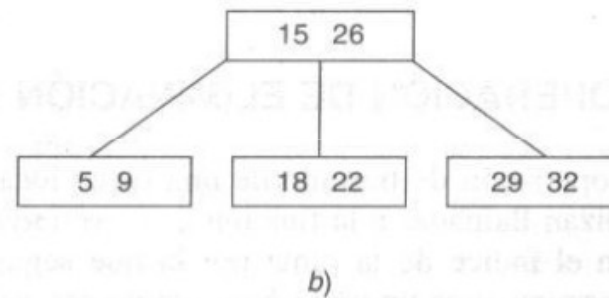
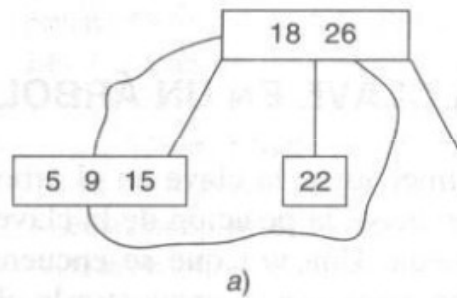


Figura 16.8. Árbol B de orden 5 para suprimir claves.

Suprime 16



Suprime 24

Suprime 22

Figura 16.9. a) Nodo hoja después de eliminar clave 24. b) Nodos después del movimiento de claves.

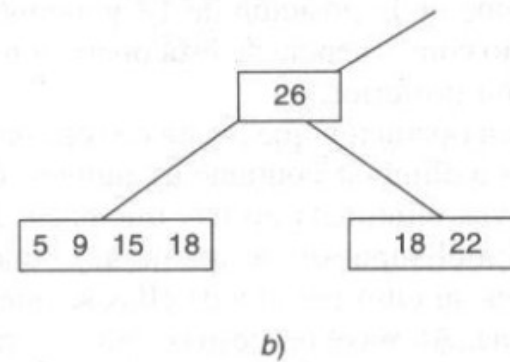
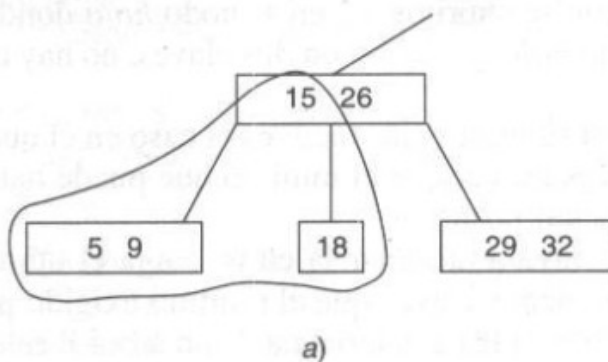


Figura 16.10. a) Combinación para restaurar árbol B. b) Árbol después de la combinación.

Ejercicios

- **Ejercicio 1**

- En un árbol B de orden 5 realizar las siguientes inserciones:
8,14,2,15,3,1,16,6,5,27,37,18,25,7,13,20,22,23,24
- Borrar la llaves: 6,7,8,16

- **Ejercicio 2**

- En un árbol B de orden 3 insertar:
5,25,10,4,8,42,1,9,11,6,32,55,18,20

- **Ejercicio 3**

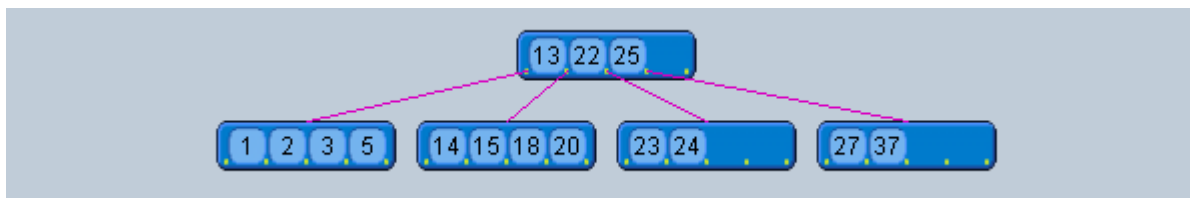
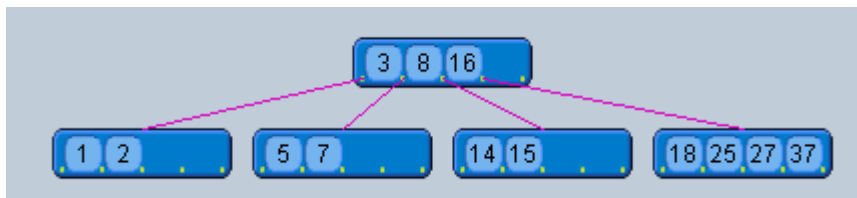
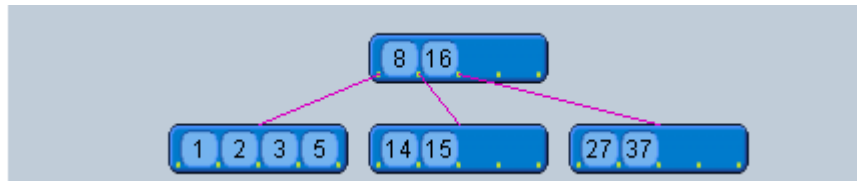
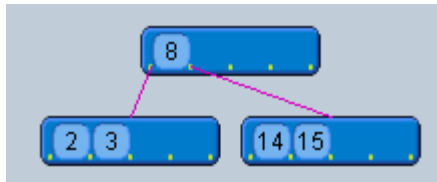
- En un árbol B de orden 5 insertar:
10,20,30,40,50,60,5,15,25,35,45,55,65
- Borrar las llaves: 15,40,55

• **Ejercicio 1**

– En un árbol B de orden 5 realizar las siguientes inserciones:

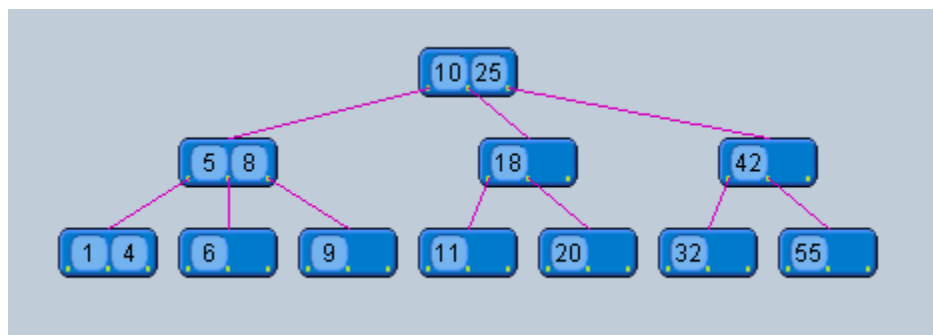
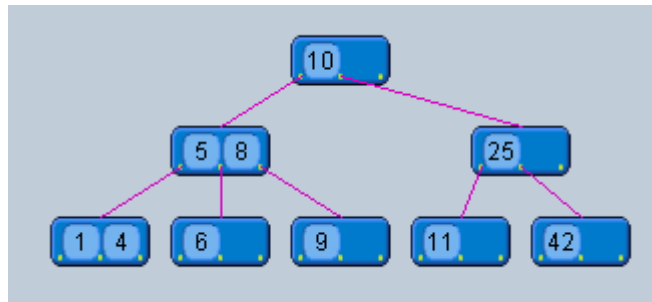
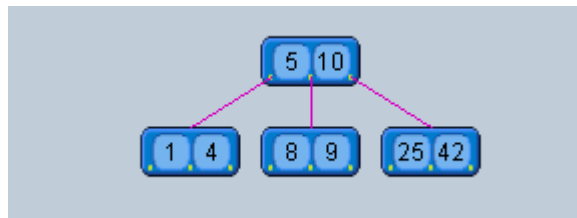
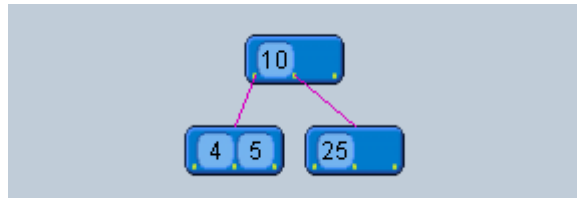
8,14,2,15,3,1,16,6,5,27,37,18,25,7,13,20,22,23,24

– Borrar la llaves: 6,7,8,16



- **Ejercicio 2**

- En un árbol B de orden 3 insertar:
5,25,10,4,8,42,1,9,11,6,32,55,18,20



Tablas de dispersión

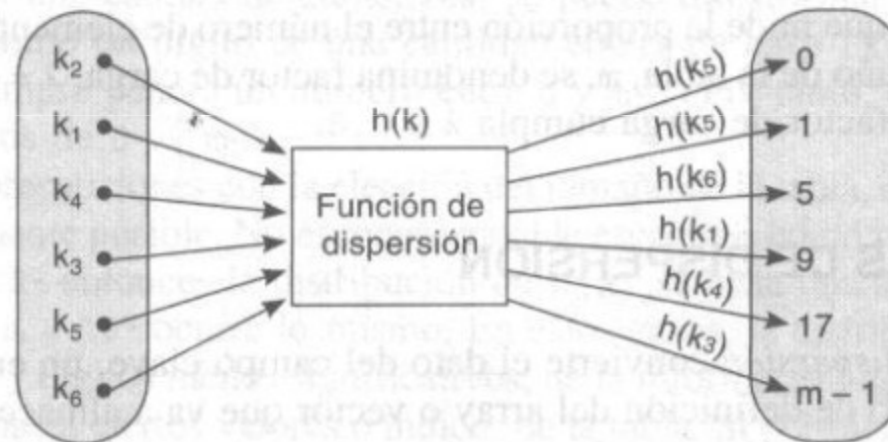
Capítulo 13, Joyanes

Para recordar

Una tabla de dispersión consta de una array, donde se almacenan los registros o elementos, y de una función hash que transforma el campo clave elegido en el rango entero del array.

Variable independiente:
campo clave

Índices de tabla



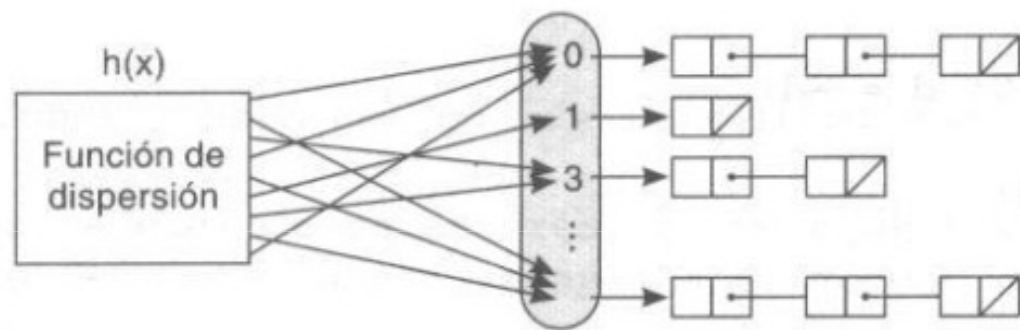
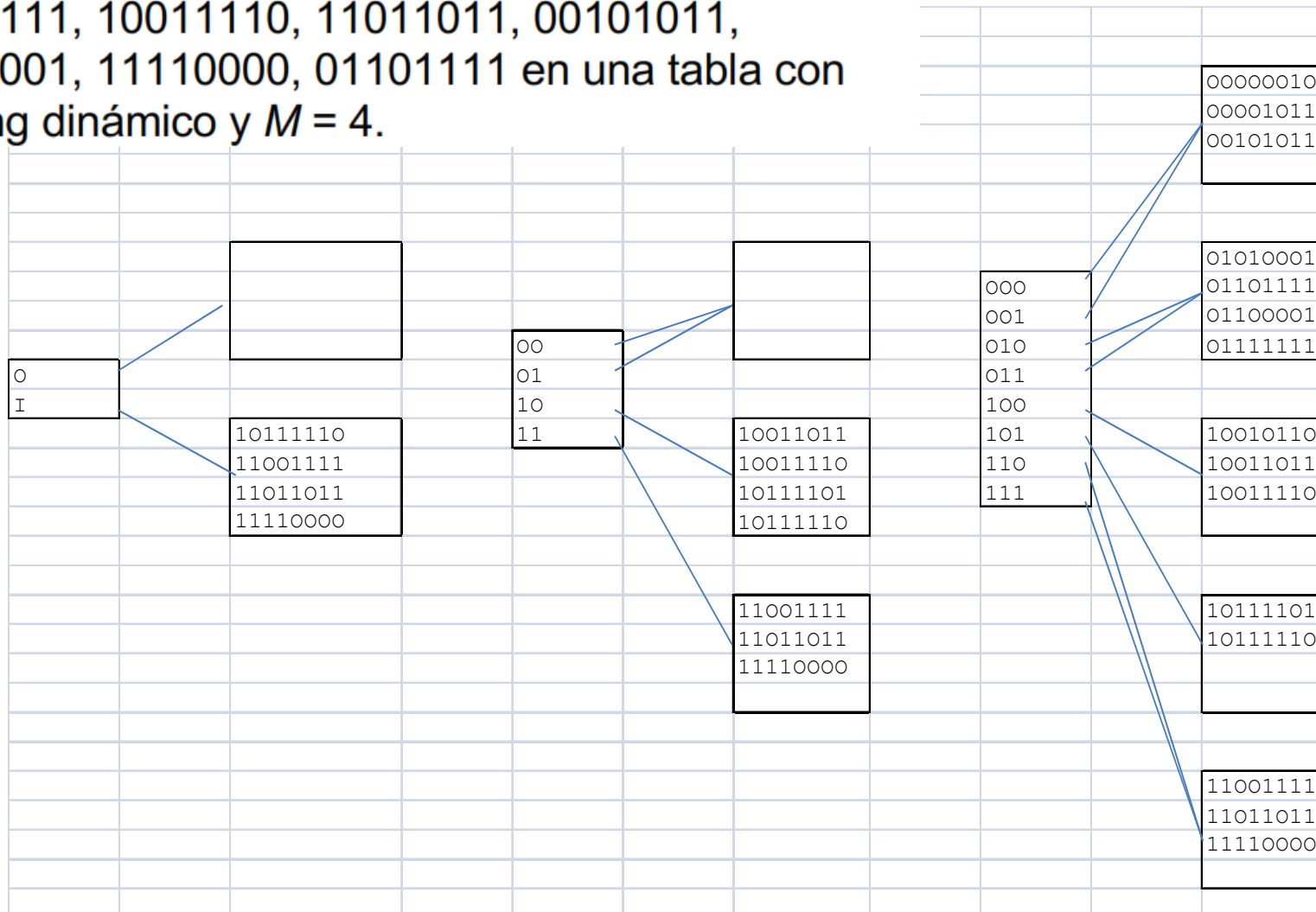


Figura 13.2. Resolución de colisiones por dispersión abierta (enlazada).

Ejercicio 2

Muestre el resultado de insertar las llaves: 10111101, 00000010, 10011011, 10111110, 01111111, 01010001, 10010110, 00001011, 11001111, 10011110, 11011011, 00101011, 01100001, 11110000, 01101111 en una tabla con hashing dinámico y $M = 4$.



Temas Examen 1

13

**TABLAS DE DISPERSIÓN,
FUNCIONES HASH**

15

**ÁRBOLES EQUILIBRADOS
DE BÚSQUEDA**

16

Árboles B

Árboles Rojo-Negro

Temas de las tareas 1-4
Programación en C
Uso básico de “make”