

Estructuras de almacenamiento de datos

Hashing

Técnica abierta (el espacio de almacenamientos se reduce o expande a partir de las bajas y altas)

- **Hashing Separado:** se crea una lista de baldes lógicos asociada a cada balde de la estructura primaria para manejar el overflow. Manteniendo el espacio físico primario.
- **Separado con crecimiento lineal:** trata de que las listas no sean demasiado largas, entonces cuando se supera cierto factor de carga preestablecido (generalmente 0,9) se crea un balde mellizo modificando el espacio físico primario.

Ejemplo de inserción (hashing separado con crecimiento lineal):

$h(x) = x \text{ MOD } 7$ (función de hashing)
 $h'(x) = x \text{ MOD } 14$ (función de reasignación)
 $r_p = 2$ (ranuras primarias)
 $r_s = 2$ (ranuras secundarias)
 $f = 0$ (frontera)
 $p_d = 0,9$ (factor de diseño)

Inicialmente:

42	29	30		32	33	6
7	43	2		25		13
0	1	2	3	4	5	6

f $N^t = 11$

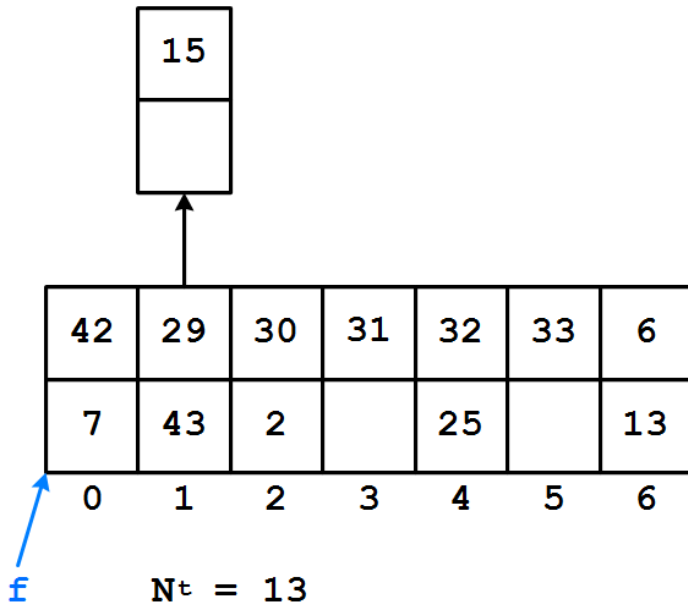
Se desea insertar (31, 15, 28, 20, 63).

Insertar(31) $\rightarrow p_t = 11/14 = 0,78 < p_d \rightarrow$ inserto normalmente
 $h(31) = 31 \text{ MOD } 7 = 3$

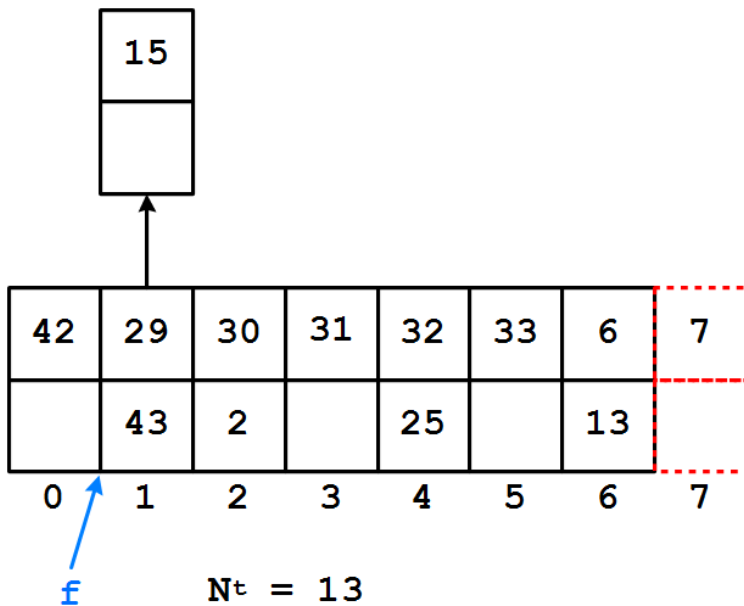
42	29	30	31	32	33	6
7	43	2		25		13
0	1	2	3	4	5	6

f $N^t = 12$

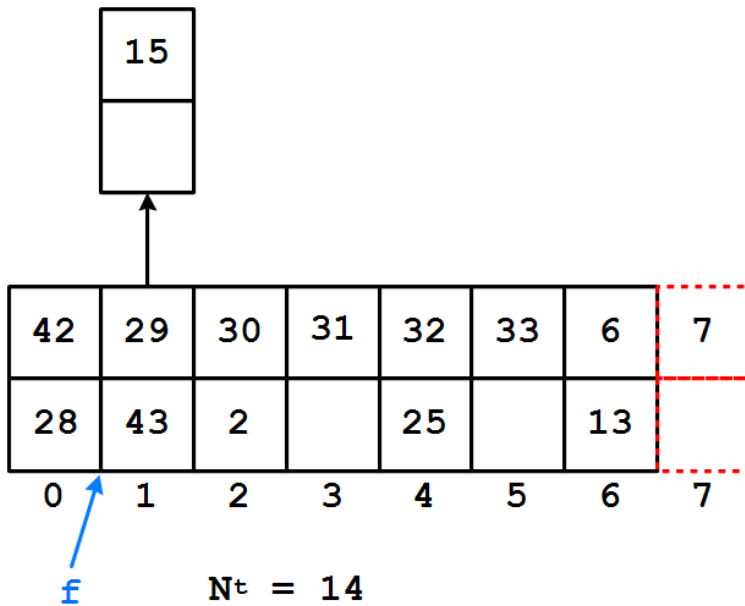
Insertar(15) $\rightarrow p_t = 12/14 = 0,85 < p_d \rightarrow$ inserto normalmente
 $h(15) = 15 \text{ MOD } 7 = 1$



Insertar(28) $\rightarrow p_t = 13/14 = 0,92 > p_d \rightarrow$ corro la frontera 1 lugar y creo un balde mellizo. (ojo acá, $13/14$ los baldes lógicos no cuentan en el p_t) reasigno todos los elementos a la izquierda de la frontera.

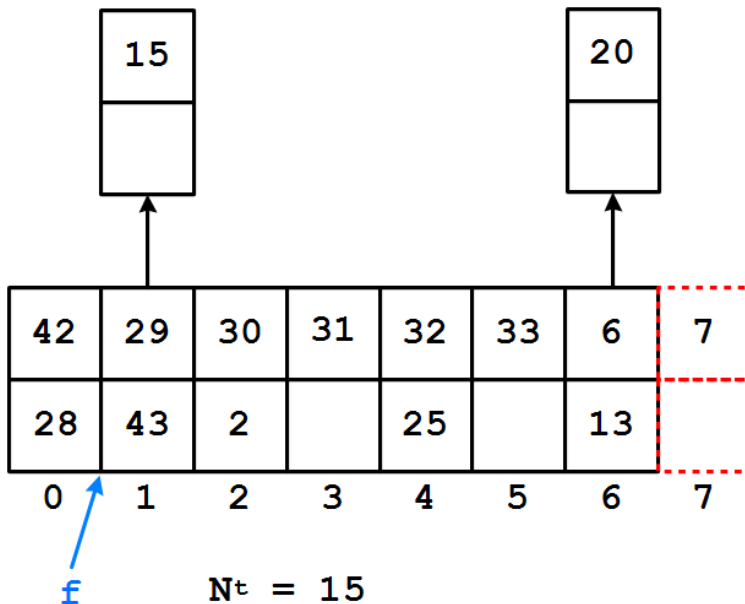


Como $h(28) = 28 \text{ MOD } 7 = 0$ está a izquierda de la frontera, debo usar la función de reasignación para saber dónde cae, entonces:
 $h'(28) = 28 \text{ MOD } 14 = 0$

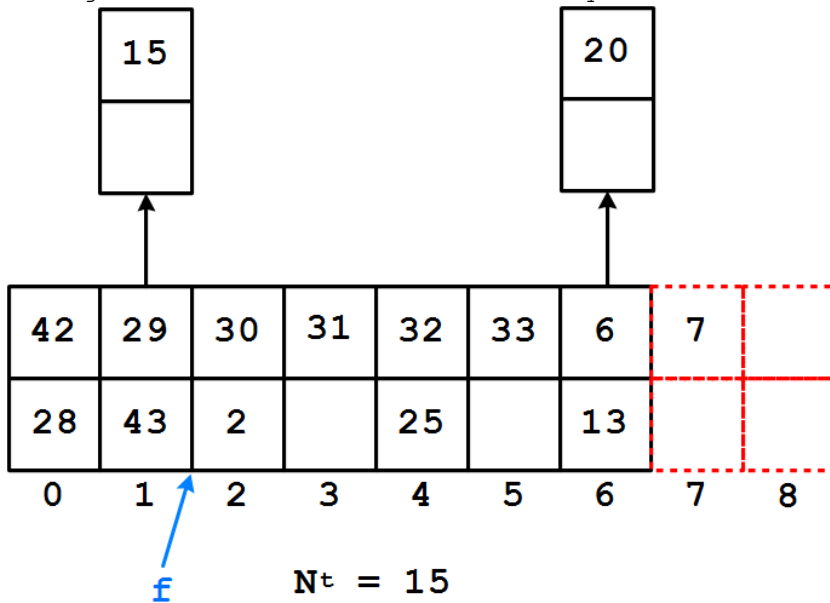


Insertar(20) $\rightarrow p_t = 14/16 = 0,87 < p_d \rightarrow$ inserto normalmente. (ojo acá, 13/16 los baldes mellizos cuentan en el p_t).

$h(20) = 20 \text{ MOD } 7 = 6$ como cae a derecha de la frontera me quedo con esta función.

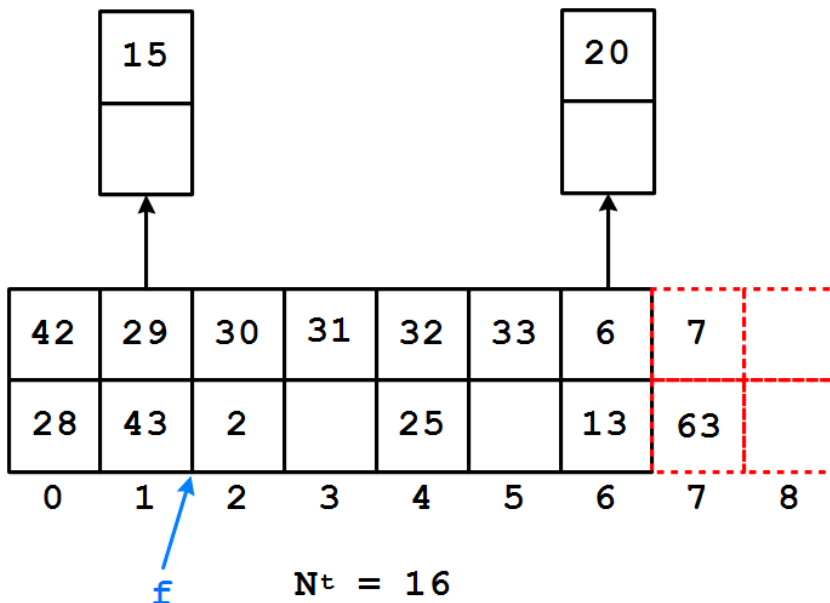


Insertar(63) $\rightarrow p_t = 15/16 = 0,93 > p_d \rightarrow$ corro la frontera 1 lugar y creo un balde mellizo y reasigno todos los elementos a la izquierda de la frontera.



En este caso al reasignar con $h'(x)$ a 15, 29, y 43 caen en el mismo lugar, entonces no pasa nada. Pero no es que no se haya hecho la reasignación, tener en cuenta que no hace falta volver a reasignar a 42 y 28, porque ya habían sido reasignados.

Como $h(63) = 63 \text{ MOD } 7 = 0$ está a izquierda de la frontera, debo usar la función de reasignación para saber dónde cae, entonces:
 $h'(63) = 63 \text{ MOD } 14 = 0$



Si se siguiesen agregando elementos hasta duplicar la completamente la estructura, se utiliza la siguiente forma de reasignación:

$h(x)$ pasa a ser $h'(x)$ o sea en nuestro caso $h(x) = x \text{ MOD } 14$

$h'(x)$ pasa a ser $x \text{ MOD } 28$

y la frontera vuelve a 0.

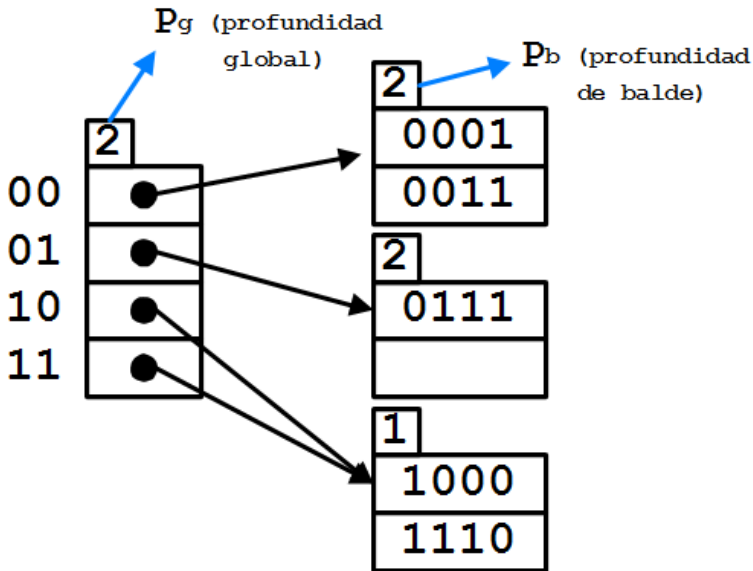
En general

$h'(x) = x \text{ MOD } 2^j M$ (con j cantidad de veces que se ha duplicado la estructura).

- **Extensible:** al igual que el hashing con crecimiento lineal modificando el espacio físico primario. Pero muy distinto en cuanto a su funcionamiento:
- La función de hash genera una secuencia de bits y se usan de a a n para direccionar a los elementos.
 - Existe un directorio con 2^n entradas. Cada entrada guarda el puntero al balde correspondiente.

Ejemplo:

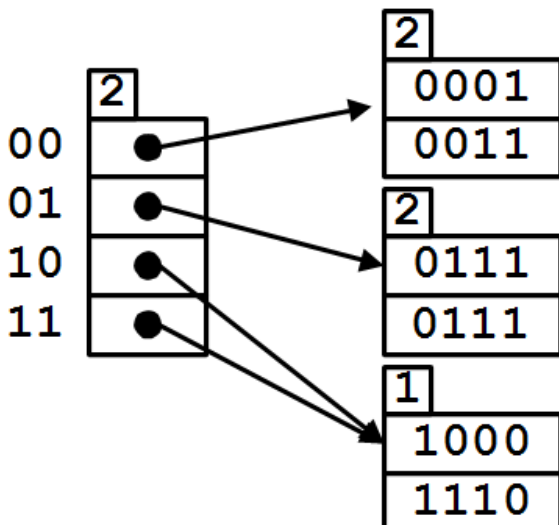
Función de hashing con secuencias de 4 bits.
Directorio de 2^2 entradas.



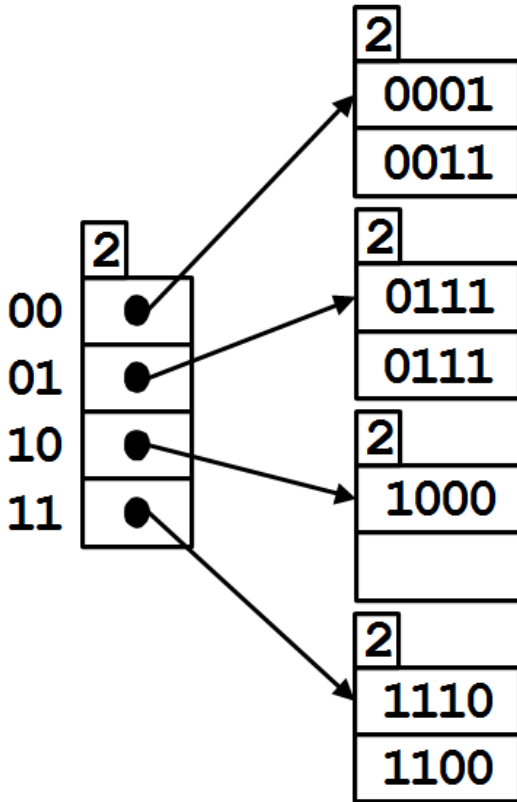
La profundidad del balde determina la cantidad de bits del directorio que utilizo para direccionar los elementos en el balde. Cuando la $P_b > P_g$ entonces se debe redimensionar el directorio.

Se desea insertar (0111, 1100, 1110, 0000).

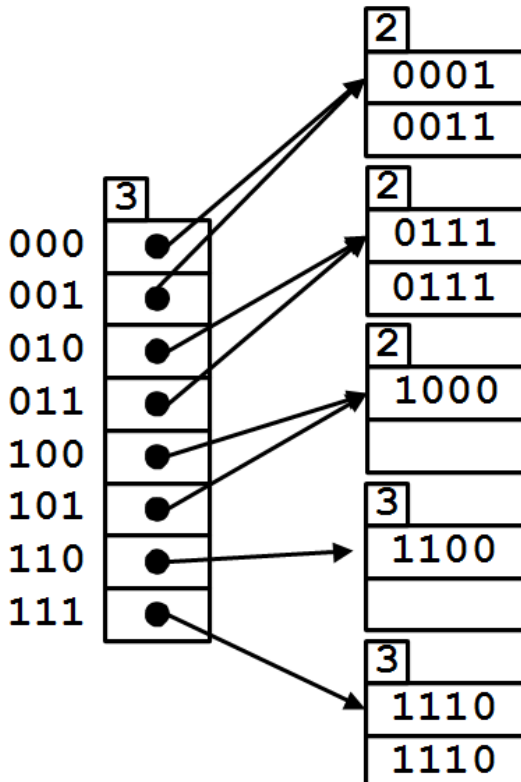
Insertar(0111), como hay lugar inserto normalmente.



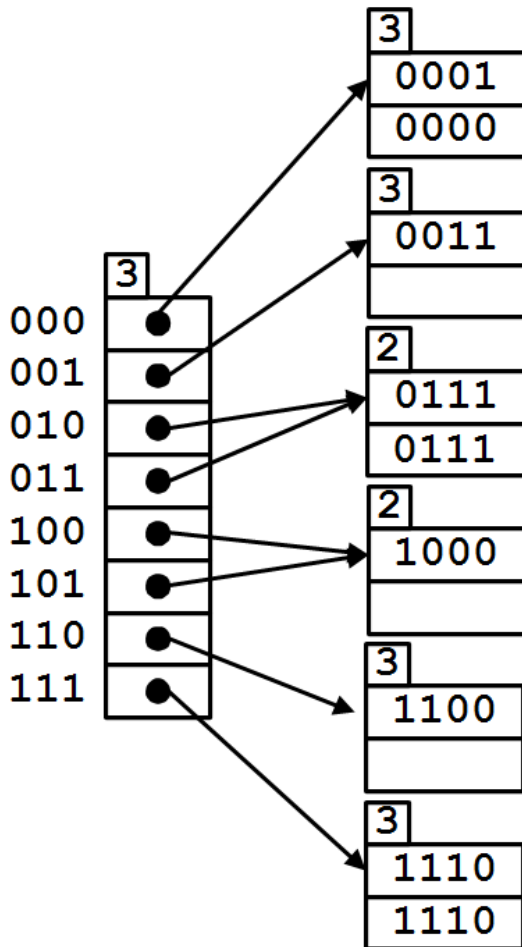
Insertar(1100) como no hay lugar se crea un nuevo balde, se actualizan las entradas del directorio y la profundidad de los baldes.



Insertar(1110) como no hay lugar se crea un nuevo balde, se actualizan las entradas del directorio la profundidad de los baldes y además como la profundidad del balde es mayor que la del directorio se debe redimensionar el mismo.



Insertar(0000) como no hay lugar se crea un nuevo balde, se actualizan las entradas del directorio y la profundidad de los baldes.



En caso de que se quiera volver a agregar el 1110 no hay lugar en el balde, y tampoco me sirve ampliar el directorio, entonces se crea un balde de desborde, el cual es apuntado por el último balde en este ejemplo. Para eliminar se siguen los pasos inversos.