

# Archivos Balanceados

## 1 INTRODUCCIÓN

---

Rudolf Bayer y Ed McCreight describieron a los árboles B en un artículo en 1972. Se mantienen balanceados en altura, con todos los nodos hoja al mismo nivel. Las operaciones de búsqueda y actualización afectan a pocos nodos. Garantizan que más de determinada proporción de la capacidad de cada nodo esté ocupada (balance de carga).

## 2 ÁRBOLES EN ARCHIVOS

---

Los nodos son bloques de organización de registros. Los registros pueden ser de longitud fija o de longitud variable. Los nodos sucesores se referencian por su posición relativa en el archivo, y las referencias son campos de control en los nodos.

Las raíces se pueden mantener siempre en una posición fija del archivo (por ejemplo la 0), para no tener que persistir la posición de la raíz en ningún lado.

Los nodos requieren campos de control generales como contador de registros contenidos y nivel del nodo (las hojas tienen el nivel más bajo).

Las hojas tienen más capacidad útil que el resto de los nodos por no requerir referencias a sucesores (en los ejemplos y en la práctica consideraremos a todos los nodos con la misma capacidad por simplicidad).

## 3 ÁRBOLES B

---

El nodo raíz es una hoja o tiene al menos dos nodos sucesores.

Cada nodo interno (no hoja) tiene un sucesor más que la cantidad de elementos que cargue, y a cada elemento se asocia un sucesor izquierdo y otro derecho.

Cada nodo, excepto el raíz, tiene ocupada al menos la mitad de su capacidad.

Los elementos de cada nodo están ordenados por un identificador.

En cada nodo, todo elemento tiene identificador mayor que el último elemento del sucesor izquierdo y menor que el primer elemento del sucesor derecho.

### 3.1 REGLAS CONVENCIONALES

El orden  $O$  de un árbol B es la máxima cantidad de hijos que puede tener cualquiera de sus nodos internos. La carga mínima de sus nodos se determina como la parte entera de la división del orden entre dos, menos uno ( $O \div 2 - 1$ ).

Para representar el estado de un árbol, se adopta como convención representar los elementos o registros mediante su clave de identificación encerrada entre paréntesis.

Por simplicidad, no se dibuja el estado de contadores de elementos ni de indicadores de tipo de nodo.

Para cada operación se indican los nodos leídos y escritos (en relación al estado previo a la operación).

Los números de nodo se deben asignar en forma coherente con el crecimiento del archivo y la reutilización de nodos libres se debe efectuar con política LIFO (último en entrar, primero en salir).

Ejemplo de Árbol B de Orden 5 en un archivo:

2: 0 (315) 1 (485) 4 (547) 5 (639) 3  
0: (148)(223)      1: (333)(390)(442)(454)      4: (508)(511)      5: (614)(633)      3: (789)(915)

## 3.2 OPERACIONES

### 3.2.1 Altas

Todo nuevo registro se inserta en una hoja.

Cuando se intenta insertar un elemento en un nodo con carga máxima, se produce un desborde o sobreflujo (overflow), que se soluciona siempre desde el nodo padre: se agrega un nodo hermano a su derecha y se redistribuye la carga entre ambos, y el elemento central de la carga sube al nodo padre, con el nuevo nodo como hijo derecho; si el orden del árbol es par, es decir, la carga máxima es un número impar de elementos, el nodo desbordado queda con un elemento más.

*Alta de registro con clave de identificación 450 en el árbol anterior:*

8: 2 (485) 7  
2: 0 (315) 1 (442) 6      7: 4 (547) 5 (639) 3  
0: (148)(223)      1: (333)(390)      6: (450)(454)      4: (508)(511)      5: (614)(633)      3: (789)(915)

L2, L1 (sobreflujo), E1, E6, sube (485) 4 a nodo 2 y hay sobreflujo, E2, E7, E8

### 3.2.2 Bajas

Así como las altas se solucionan agregando el nuevo elemento en una hoja, las bajas se solucionan eliminando un elemento de una hoja. Para eliminar un elemento en un nodo que no es hoja, se lo sustituye por el elemento siguiente, que es el primero de la primera hoja del subárbol derecho del elemento a eliminar.

*Baja de registro con clave de identificación 485 (como el orden del árbol es 5, la capacidad mínima de sus nodos es  $5/2-1=1$ ):*

8: 2 (508) 7  
2: 0 (315) 1 (442) 6      7: 4 (547) 5 (639) 3  
0: (148)(223)      1: (333)(390)      6: (450)(454)      4: (511)      5: (614)(633)      3: (789)(915)

L8, L7, L4 (baja de 508), E4, reemplazo de 485 por 508 en 8, E8

Cuando se elimina un elemento de un nodo con carga mínima, se produce un subflujo (underflow), que se soluciona siempre desde el nodo padre: se lee a su hermano izquierdo (si no tuviera, excepcionalmente al derecho), y si éste tiene carga mínima, el elemento puente entre ambos nodos en el padre más la carga del nodo con subflujo pasan al hermano izquierdo (fusión de cargas y liberación del nodo derecho), si no, se

*Baja de registro con clave de identificación 511:*

3: (789)(915)

*Baja de registro con clave de identificación 614:*

3: (789)(915)

**Nodos libres: 5**

1. Dado el árbol B que se muestra abajo, de orden 6, realizar las siguientes operaciones en secuencia: alta 740, baja 995. Justificar cada operación indicando nodos leídos y escritos.

4: (992)(995)

2. Dado el siguiente árbol B de orden 8, realizar las operaciones que se indican a continuación, cada una sobre el resultado de la operación anterior, indicando la secuencia de nodos leídos y escritos: alta 50, alta 512, baja 162.

2: 0 (162) 1 (251) 3 (694) 4 (964) 5

0: (3)(47)(108)(117)(124)(132)(157)

1: (214)(227)(238)

3: (387)(506)(509)(535)(627)(643)(644)

4: (737)(740)(812)

5: (976)(993)(998)

3. Dado el árbol B que se muestra a continuación, de orden 6, mostrar los estados sucesivos completos al efectuarle las siguientes operaciones en el orden en que se especifican: alta 760, baja 755 y baja 964. Indicar la secuencia de nodos leídos y escritos en cada operación.

2: 0 (83) 1 (379) 3 (755) 4

0: (34)( 80)

1: (139)(221)

3: (522)(652)(695)

4: (771)(849)(853)(907)(964)

4. Dado el siguiente árbol B de orden 6, dibuje los árboles resultantes para la siguiente secuencia de operaciones, indicando los nodos leídos y escritos en cada una: baja 900, baja 897, baja 762, alta 20.

9: 2 (801) 8

2: 0 (184) 1 (387) 3 (611) 4	8: 5 (880) 6 (910) 7	
0: (10)(19)(40)(52)(140)	5: (838)(847)(858)	
1: (199)(200)(238)(266)(278)	6: (897)(900)	
3: (461)(463)(489)(569)(570)	7: (935)(1003)	
4: (750)(762)		

## 4 ÁRBOLES B+

Se diferencia de los árboles B en que los registros sólo se almacenan en los nodos hoja, que están encadenados entre sí para permitir su recuperación secuencial (por eso las hojas también se denominan nodos de secuencia). Los nodos internos o índice, almacenan sólo identificadores que sirven de guía para llegar a las hojas, conformando un índice B a las hojas.

### 4.1 REGLAS CONVENCIONALES

El orden  $O$  de un árbol B+ es la máxima cantidad de hijos que puede tener cualquiera de sus nodos internos. La carga mínima de sus nodos se determina como la parte entera de la división del orden entre dos, menos uno ( $O \div 2 - 1$ ).

Para representar el estado de un árbol, se adopta como convención representar los elementos o registros de sus hojas mediante su clave de identificación encerrada entre paréntesis. En los nodos internos los elementos son claves de identificación simples.

Por simplicidad, no se dibuja el estado de contadores de elementos ni de indicadores de tipo de nodo.

Para cada operación se indican los nodos leídos y escritos (en relación al estado previo a la operación).

Los números de nodo se deben asignar en forma coherente con el crecimiento del archivo y la reutilización de nodos libres se debe efectuar con política LIFO (último en entrar, primero en salir).

Ejemplo de Árbol B+ de Orden 4 en un archivo:

2: 0 (432) 3 (658) 1 (724) 4

0: 153, 254   3	3: 432, 508, 620   1	1: 658, 702   4	4: 905   -1
-----------------	----------------------	-----------------	-------------

### 4.2 OPERACIONES

#### 4.2.1 Altas

Todo nuevo registro se inserta en una hoja.

Cuando se intenta insertar un elemento en un nodo con carga máxima, se produce un desborde o sobreflujo (overflow), que se soluciona siempre desde el nodo padre.

Si se desborda una hoja se agrega un nodo hermano a su derecha y se redistribuye la carga del nodo desbordado más el registro nuevo entre él y el nodo nuevo a su derecha, y en el nodo padre se agrega una copia de la clave del primer registro del nodo nuevo con el número del nuevo nodo como referencia al hijo derecho; si el orden del árbol es impar, es decir, la carga del nodo desbordado más el nuevo registro es un número impar de registros, el nodo desbordado queda con un elemento más.

Si se desborda un nodo interno, el problema se resuelve igual que en los árboles B.

*Alta de registro con clave de identificación 500 en el árbol anterior:*

7: 2(658)6				
2: 0 (432) 3 (508) 5			6: 1 (724) 4	
0: 153, 254   3	3: 432, <u>500</u>   5	5: 508, 620   1	1: 658, 702   4	4: 905   -1

L2, L3 (desborde y solución del mismo desde nodo 2), E3, E5, inserción de clave 508 con referencia a hijo derecho 3 en nodo 2 (desborde y solución del mismo desde fuera del árbol), E2, E6, E7.

#### 4.2.2 Bajas

Las bajas siempre implican la eliminación de un registro en una hoja, por lo que para realizarlas siempre se debe llegar a una hoja, aunque se encuentre la clave de identificación del registro que se busca en un nodo interno.

Cuando se borra el primer registro de una hoja, que en el nodo padre tiene una copia de su clave de identificación, sin que se produzca un balanceo o fusión con su hermano izquierdo, no se actualiza la clave de identificación de ese registro en el nodo padre (para evitar la reescritura de este nodo).

*Baja de registro con clave de identificación 658:*

7: 2(658)6				
2: 0 (432) 3 (508) 5			6: 1 (724) 4	
0: 153, 254   3	3: 432, 500   5	5: 508, 620   1	1: <del>658</del> , 702   4	4: 905   -1

L7 (se encuentra clave de registro a eliminar pero aquí no está el registro), L6, L1 (eliminación de registro), E1  
No se reemplaza la clave 658 por la 702 en el nodo 7 para ahorrar la operación de escritura del nodo 7

*Baja de registro con clave de identificación 702:*

7: 2 (508) 6				
2: 0 (432) 3			6: 5 (658) 1	
0: 153, 254   3	3: 432, 500   5	5: 508, 620   1	1: <del>658, 702</del> , 905   -1	

L7, L6, L1, al eliminar el registro se produce subflujo que se soluciona desde el nodo 6, L4 (se lee su hermano derecho porque izquierdo no tiene), como el nodo 4 tiene carga mínima su carga debe pasarse a su hermano izquierdo (fusión), E1, al eliminar la clave 724 con la referencia al nodo 4 se produce subflujo en nodo 6 que se soluciona desde nodo 7, L2, como el nodo 2 no tiene carga mínima se balancean 2 y 6, E2, E6, E7.

**Nodos Libres: 4**

**Nota:** observar que debe liberarse el nodo 4, ya que de liberarse el 1 habría que cambiar el puntero al siguiente del 5, que no se ha leído.

### 4.3 EJERCICIOS

5. Dado el árbol B+ que se muestra abajo, con capacidad para 5 registros en nodos hoja y 5 claves en nodos internos, realizar las siguientes operaciones en secuencia: alta 125, baja 952, baja 504. Justificar cada operación indicando nodos leídos y escritos.

2: 0 (504) 1 (685) 3 (937) 4

0: (69)(192)(262)(447)(500) 1 1: (504)(595) 3 3: (685)(700)(715)(883)(886) 4 4: (937)(952) -1

6. Dado el siguiente árbol B+ de nodos internos con capacidad para 4 claves y hojas con capacidad para 4 registros (carga mínima 1 en ambos casos), realizar las operaciones que se indican a continuación, cada una sobre el resultado de la operación anterior, indicando la secuencia de nodos leídos y escritos: alta 100, baja 599, baja 636.

2: 0 (289) 1 (477) 3 (599) 4

0: (32)(50)(262)(270) 1 1: (289)(473) 3 3: (477)(534)(545) 4 4: (599)(636) -1

7. Dado el árbol B+ que se muestra abajo, con capacidad para 5 registros en nodos hoja y 5 claves en nodos internos, realizar las siguientes operaciones en secuencia: alta 18, baja 48, baja 37. Justificar cada operación indicando nodos leídos y escritos.

9: 2 (37) 8

2: 0 (8) 1 (15) 3 (28) 4

8: 7 (48) 5 (105) 6

0: (2)(6) 1

7: (37)(40)(45) 5

1: (8)(9)(11) 3

5: (48)(50) 6

3: (15)(17)(22)(24)(26) 4

6: (105)(111)(196) -1

4: (28)(30)(35) 7

8. Dado el siguiente árbol B+ de nodos internos con capacidad para 6 claves y hojas con capacidad para 6 registros (carga mínima 3 en ambos casos), realizar las operaciones que se indican a continuación, cada una sobre el resultado de la operación anterior, indicando la secuencia de nodos leídos y escritos: baja 178, baja 90, alta 200.

10: 2 (185) 9

2: 0 (93) 1 (116) 3 (171) 4

9: 5 (218) 6 (564) 7 (678) 8

0: ( 18)( 81)( 90) 1

5: (185)(187)(197)(198)(203)(207) 6

1: (93)(99)(101) 3

6: (218)(230)(309)(509) 7

3: (116)(153)(155)(170) 4

7: (564)(589)(604) 8

4: (171)(178)(180) 5

8: (678)(769)(801)(842) -1

9. Dada las siguientes definiciones para nodos de un árbol B+, calcular los valores de CMxI, CMxH y CR (una posibilidad es codificar un programa con estas definiciones que sólo imprima en pantalla los valores de estas constantes) y analizar la implicancia de que CMxI sea distinta que CMxH:

Const

TN=1024; {tamaño de nodo en bytes}

Type

TClave=Longword; {tipo de clave de identificación de elementos a organizar en el árbol}

TElemento = packed Record {registro de persona}

dni: TClave;  
apellido: String[20];  
nombre: String[20];  
fNac: Longword  
end;

TRefBMas = Integer; {tipo de referencias a nodos sucesores (números relativos de nodo)}

Const

CMxI=(TN-2-SizeOf(TRefBMas)) div (SizeOf(TClave)+SizeOf(TRefBMas)); {cantidad máxima de **claves** en un nodo interno}

CR=(TN-2-SizeOf(TRefBMas)) mod (SizeOf(TClave)+SizeOf(TRefBMas)); {cantidad de bytes de relleno}

CMxH=(TN-2-SizeOf(TRefBMas)) div SizeOf(TElemento); {cantidad máxima de **registros** en un nodo hoja}

Type

TNodoBMas = packed Record {nodo de arbol B+}

cont: Byte; {contador de claves o elementos contenidos efectivamente}

Case tipo: Char of {tipo de nodo ('i' interno u 'h' oja)}

'i': (c: Array[1..CMxI] of TClave; {claves en nodo interno}

suc: Array[0..CMxI] of TRefBMas; {referencias a nodos sucesores}

relleno: Array[1..CR] of Byte);{relleno para completar el tamaño de nodo objetivo}

'h': (e: Array[1..CMxH] of TElemento; {registros de nodo hoja – en las hojas no hacen falta referencias a nodos sucesores}

sig: TRefBMas) {referencia a siguiente hoja o -1 si fuera la última}

end;