6- Orden 5, política a izquierda en caso de underflow

2: 0 (320) 1 (490) 4 (555) 5 (641) 3

0: (13)(153) 1: (325)(341)(431)(460) 4: (500)(507) 5: (608)(633) 3: (790)(923)

+445 -> Produce overflow en 1, 325 341 431 445 460, se genera el nuevo nodo 6, de estos se sube el 431 al padre, se vuelve a producir un overflow (ahora en nodo 2), 320 431 490 555 641, se crea el nodo 7, 490 sube como raíz en un nuevo nodo que sería el 8.

L2, L2,E1,E6,E2,E7,E8

8: 2 (490) 7

2: 0 (320) 1 (431) 6 7: 4 (555) 5 (641) 3

0: (13)(153) 1: (325)(341) 6:(445)(460) 4: (500)(507) 5: (608)(633) 3: (790)(923)

-490 -> Para remover el 490 se busca el menor de su subárbol derecho, que sería 500

L8,L7,L4,E8

8: 2 (500) 7

2: 0 (320) 1 (431) 6 7: 4 (555) 5 (641) 3

0: (13)(153) 1: (325)(341) 6:(445)(460) 4: (507) 5: (608)(633) 3: (790)(923)

-507 -> Se produce underflow, busca en el adyacente izquierdo, como no tiene busca en el derecho, 555 608 633, se divide equitativamente y sube como padre el 608 al nodo 7.

L8, L7, L4, L5, E4, E5, E7

8: 2 (500) 7

2: 0 (320) 1 (431) 6 7: 4 (608) 5 (641) 3

0: (13)(153) 1: (325)(341) 6:(445)(460) 4: (555) 5: (633) 3: (790)(923)

-608 -> Se busca el menor del subárbol derecho de 608, que es 633, se produce underflow en 5, se hace unión del nodo 4 y 5 en el nodo 4 (liberando el nodo 5 que se guarda como espacio libre para alocar el próximo elemento a ingresar).

L8, L7, L5, E7, L4, E4, E7

8: 2 (500) 7

2: 0 (320) 1 (431) 6 7: 4 (641) 3

0: (13)(153) 1: (325)(341) 6:(445)(460) 4: (555)(633) 3: (790)(923)

8- Árbol B+ de orden 4, con política de underflow a derecha.

4: 0 (340) 1 (400) 2 (500) 3

0: (11)(50)(77) 1 1: (340)(350)(360) 2 2: (400)(410)(420) 3 3: (520)(530) -1

+150 -> Produce overflow en el nodo 0, 11 50 77 150, se crea nuevo nodo 5, sube a raíz 77 como padre del nodo 0 y 5, se produce overflow en la raíz, 77 340 400 500, se crea el nuevo nodo 6, sube como raíz a el nuevo nodo 7 el 400, siendo padre del nodo 4 y 6.

L4, L0, E0, E5, E4, E6, E7

7: 4 (400) 6

4: 0 (77)5 (340)1 6: 2 (500) 3

0: (11)(50) 5:(77)(150) 1 1: (340)(350)(360) 2 2: (400)(410)(420) 3 3: (520)(530) -1

-500 -> No se puede eliminar la clave porque no es una hoja

9- Árbol B con orden 4, ante disparidad de carga, el nodo izquierdo queda más cargado y la política de underflow es a la derecha.

2: 0 (56) 1

0: (23)(34)(40) 1: (67) (89) (102)

+2 -> Se produce overflow en el nodo 0, 2 23 34 40, se crea el nuevo nodo 3 y se sube a raiz la clave 34.

L2, L0, E0, E2, E3

2: 0 (34) 3 (56) 1

0: (2)(23) 3:(40) 1: (67) (89) (102)

+10 -> Se agrega al nodo 0 sin problemas

L2, L0, E0

2: 0 (34) 3 (56) 1

0: (2)(10)(23) 3:(40) 1: (67) (89) (102) -1

+95 -> Se produce overflow en el nodo 1, 67 89 95 102, se crea el nuevo nodo 4 y sube a raiz el 95.

L2, L1, E1, E4, E2

2: 0 (34) 3 (56) 1 (95) 4

0: (2)(10)(23) 3:(40) 1: (67)(89) 4:(102)

+60 -> Se agrega correctamente al nodo 1

L2, L1, E1

2: 0 (34) 3 (56) 1 (95) 4

0: (2)(10)(23)3 3:(40) 1: (60)(67)(89)4 4:(102) -1

-40 -> Se produce underflow en el nodo 3, se trata de redistribuir siguiendo la política de derecha, como no tiene nodo adyacente a su derecha solo queda hacerlo con el de la izquierda, 2 10 23 34, para redistribuir queda 23 como padre de el nodo 0 y 3.

L2, L3, L0, E0, E3, E2

2: 0 (23) 3 (56) 1 (95) 4

0: (2)(10)3 3:(34) 1: (60)(67)(89)4 4:(102) -1

-102 -> Se produce underflow en el nodo 4, se trata de redistribuir siguiendo la política de derecha, como no tiene nodo adyacente a su derecha solo queda hacerlo con el de la izquierda, 60 67 89 95, para redistribuir queda 89 como padre de el nodo 1 y 4.

L2, L4, L1, E1, E4, E2

2: 0 (23) 3 (56) 1 (89) 4

0: (2)(10)3 3:(34) 1: (60)(67)4 4:(95) -1

10- Árbol B de orden 6, ante disparidad de carga el nodo izquierdo queda más cargado y con política de underflow a derecha o izquierda.

0: (34) (56) (78) (100) (176)

+15 -> Se produce overflow en el nodo 0, 15 34 56 78 100 170, se crea el nodo 1 y el nodo 2 sube a raíz como padre del nodo 0 y 1.

L0, E0, E1, E2

2: 0 (78)  
 0: (15) (34)(56) 1: (100) (176) -1

+71 -> Se agrega correctamente al nodo 0

L2, L0, E0

2: 0 (78)  
 0: (15)(34)(56)(71) 1: (100) (176) -1

+3 -> Se agrega correctamente al nodo 0

L2, L0, E0

2: 0 (78)  
 0: (3)(15)(34)(56)(71) 1: (100) (176) -1

+48 -> Se produce overflow en el nodo 0, 3 15 34 48 56 71, se crea el nuevo nodo 3 y se sube a raíz la clave 48.

L2, L0, E0, E3, E2

2: 0(48)3 (78)1  
 0: (3)(15)(34)3 3:(56)(71) 1: (100) (176) -1

-56 -> Se produce underflow, se aplica la política de underflow de derecha o izquierda, como no tiene nodo adyacente a su derecha se fija en el de la izquierda, se intenta redistribuir 3 15 34 48 71, se sube como padre de nodo 0 y 3 al 34.

L2, L3, L0, E0, E3, E2

2: 0(34)3 (78)1  
 0: (3)(15)3 3:(48)(71) 1: (100) (176) -1

-71 -> Se elimina 71, se produce underflow en el nodo 3, se busca redistribuir con el de la derecha, como no hay se busca con el de la izquierda, como no es posible redistribuir se produce una unión, liberando al nodo 3 y dejando 0 15 34 48 en el nodo 3 y 48 como padre.

L2, L3, L0, E3, E2

2: 0(78)1  
 0: (3)(15)(34)(48) 1: (100) (176) -1

11- Árbol B con orden 5, ante disparidad de carga el nodo izquierdo queda más cargado, política de resolución a derecha.

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)

+450 -> Se produce overflow en el nodo 1, 333 390 442 450 454 crea nuevo nodo 6 y sube el 442 a raíz, se vuelve a producir overflow, 315 442 485 547 639, se crea el nuevo nodo 7 y se sube 485 como raíz en el nuevo nodo 8 y padre de los nodos 2 y 7.

L2, L1, E1, E6, E2, E7, E8

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)

-485 -> Se busca en su subárbol derecho la clave más chica y se reemplaza por 485

L8, L7, L4, E8

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)

-511 -> Se produce underflow, se busca en su adyacente derecho para hacer la redistribución, 547 614 633, nos queda 614 como padre de los nodos 4 y 5

L8, L7, L4, L5, E4, E5, E7

8: 2(508)7

2: 0 (315) 1 (442) 6 7: 4 (614) 5 (639) 3

0: (148)(223) 1: (333)(390) 6:(450)(454) 4: (547) 5: (633) 3: (789)(915)

-614 -> Se busca en el subárbol derecho el menor y se toma, se produce underflow en el nodo 5, aplicamos política a derecha, 639 789 915, queda 789 como padre del nodo 5 y 3

L8, L7, L5, E7, L3, E5, E3, E7

8: 2(508)7

2: 0 (315) 1 (442) 6 7: 4 (633) 5 (789) 3

0: (148)(223) 1: (333)(390) 6:(450)(454) 4: (547) 5: (639) 3: (915)

12- Árbol B con orden 5, ante disparidad de carga el nodo izquierdo queda más cargado y con política de underflows a izquierda.

2: 0 (56) 1

0: (10)(35)(45)(50) 1: (66) (70) (88)(120)

+15 -> Se produce overflow en el nodo 0, 10 15 35 45 50, se crea el nodo 3 y 35 sube a raíz como padre de los nodos 0 y 3.

L2, L0, E0, E3, E2

2: 0 (35) 3 (56) 1

0: (10)(15) 3:(45)(50) 1: (66) (70) (88)(120)

-45 -> Se elimina el 45 sin problemas

L2, L3, E3

2: 0 (35) 3 (56) 1

0: (10)(15) 3:(50) 1: (66) (70) (88)(120)

-50 -> Se produce underflow en el nodo 3, se aplica la política de izquierda, 10 15 35, queda 15 como padre en raíz de los nodos 0 y 3

L2, L3, L0, E0, E3, E2

2: 0 (15) 3 (56) 1

0: (10) 3:(35) 1: (66) (70) (88)(120)

13- Árbol B con orden 4, ante disparidad de carga el nodo izquierdo queda más cargado y con política de underflows a izquierda o derecha.

+50 -> Se agrega sin problemas

E0

0: (50)

+70 -> Se agrega sin problemas y se acomoda el orden

L0, E0

0: (50)(70)

+40 -> Se agrega sin problemas y se acomoda el orden

L0, E0

0: (40)(50)(70)

+15 -> Se produce underflow en el nodo 0, 15 40 50 70, se crea nodo 1 y se sube a raiz como padre de 0 y 1 a la clave 50.

L0, E0, E1, E2

2: 0(50)1

0:(15)(40) 1:(70)

+90 -> Se agrega sin inconvenientes

L2, L1, E1

2: 0(50)1

0:(15)(40) 1:(70)(90)

+120 -> Se agrega sin inconvenientes

L2, L1, E1

2: 0(50)1

0:(15)(40) 1:(70)(90)(120)

+115 -> Se produce overflow en el nodo 1, 70 90 115 120, se crea nuevo nodo 3, 115 sube a la raiz y queda como padre de los nodos 1 y 3.

L2, L1, E1, E3, E2

2: 0(50)1 (115)3

0:(15)(40) 1:(70)(90) 3:(120)

+45 -> Se agrega sin inconvenientes

L2, L0, E0

2: 0(50)1 (115)3

0:(15)(40)(45) 1:(70)(90) 3:(120)

+30 -> Se produce overflow en el nodo 0, 15 30 40 45, se crea nuevo nodo 4, 40 sube a la raiz y queda como padre de los nodos 0 y 4.

L2, L0, E0, E4, E2

2: 0(40)4 (50)1 (115)3

0:(15)(30) 4:(45) 1:(70)(90) 3:(120)

+100 ->Se agrega sin inconvenientes

L2, L1, E1

2: 0(40)4 (50)1 (115)3

0:(15)(30) 4:(45) 1:(70)(90)(100) 3:(120)

+112 -> Se produce overflow en el nodo 1, 70 90 100 112, 100, se crea el nuevo nodo 5, 100 sube a raiz como padre de los nodos 1 y 5. Se produce overflow en el nodo 2, 40 50 100 115, se crea el nuevo nodo 6, sube a raíz la clave 100 (en un nodo nuevo que sería el 7) como padre de los nodos 2 y 6.

L2, L1, E1, E5, E2, E6, E7

7:(100)

2: 0(40)4 (50)1 6: 5(115)3

0:(15)(30) 4:(45) 1:(70)(90) 5:(112) 3:(120)

+77 -> Se agrega sin inconvenientes

L7, L2, L1, E1

7:(100)

2: 0(40)4 (50)1 6: 5(115)3

0:(15)(30) 4:(45) 1:(70)(77)(90) 5:(112) 3:(120)

-45 -> Se produce un underflow, aplica la política y se redistribuye con el de la izquierda, 15 30 40, 30 sube al nodo 2 quedando como padre de 0 y 4.

L7, L2, L4, L0, E0, E4, E2

7:(100)

2: 0(30)4 (50)1 6: 5(115)3

0:(15) 4:(40) 1:(70)(77)(90) 5:(112) 3:(120)

-40 -> Se produce underflow en el nodo 4, se fija si puede redistribuir con el de la izquierda como no puede se fija con el de la derecha, 50 70 77 90, 77 sube al nodo 2 como padre del nodo 4 y 1.

L7, L2, L4, L0, L1, E4, E1, E2

7:(100)

2: 0(30)4 (77)1 6: 5(115)3

0:(15) 4:(50)(70) 1:(90) 5:(112) 3:(120)

-50 -> Se elimina 50 sin mayores inconvenientes

L7, L2, L4, E4

7:(100)

2: 0(30)4 (77)1 6: 5(115)3

0:(15) 4:(70) 1:(90) 5:(112) 3:(120)

-90 -> Se elimina 90, produce underflow en el nodo 1, se busca redistribuir con el de la izquierda, no puede, busca con el de la derecha pero no hay, entonces se fusiona con el nodo adyacente izquierdo, quedando 70 77 en el nodo 1, liberando el nodo 4, quedando como hijo derecho del nodo 2.

nodo libre: 1

L7, L2, L1, L4, E1, E2

7:(100)

2: 0(30)1 6: 5(115)3

0:(15) 1:(70)( 77) 5:(112) 3:(120)

-100 -> Se reemplaza el 100 por su menor clave del subárbol derecho, se produce underflow en el nodo 5, se busca redistribuir con el nodo adyacente izquierdo, como no hay busca el derecho, como no puede redistribuir hace una fusión entre 115 y 120 en el nodo 5, liberando el nodo 3. Se produce underflow en 6, se busca redistribuir con el adyacente izquierdo, como no puede se fusiona todo en el nodo 2, liberando el nodo 6 y 7

Nodo libre: 7 6 3 1

L7, L6, L5, E7, L3, E5, L2, E2

2: 0(30)1:( 112)

0:(15) 1:(70)( 77) 5:(120) (115)

14- Árbol B con orden 5, ante disparidad de carga el nodo izquierdo queda más cargado y con política de underflows a izquierda

+80 -> Se agrega sin problemas

E0

0: (80)

+50 -> Se agrega sin problemas y se acomoda el orden

L0, E0

0: (50)(80)

+70 -> Se agrega sin problemas y se acomoda el orden

L0, E0

0: (50)(70)(80)

+120 -> Se agrega sin problemas

L0, E0

0: (50)(70)(80)(120)

+23 -> Se produce overflow, 23 50 70 80 120, se crea el nodo 1, el 70 sube a raíz como padre del nodo 0 y 1 creando el nuevo nodo 0.

L0, E0, E1, E2

2: 0(70)1

0: (23)(50) 1: (80)(120)

+52 -> Se agrega sin problemas

L2, L0, E0

2: 0(70)1

0: (23)(50)(52) 1: (80)(120)

+59 -> Se agrega sin problemas

L2, L0, E0

2: 0(70)1

0: (23)(50)(52)(59) 1: (80)(120)

+65 -> Se produce overflow en el nodo 0, 23 50 52 59 65, se crea el nuevo nodo 3 y sube 52 a raiz como padre del nodo 0 y 3.

L2, L0, E0, E3, E2

2: 0(52)3 (70)1

0: (23)(50) 3:(59)(65) 1: (80)(120)

+30 -> Se agrega sin problemas

L2, L0, E0

2: 0(52)3 (70)1

0: (23)(30)(50) 3:(59)(65) 1: (80)(120)

+40 -> Se agrega sin problemas

L2, L0, E0

2: 0(52)3 (70)1

0: (23)(30)(40)(50) 3:(59)(65) 1: (80)(120)

+45 -> Se produce overflow en el nodo 0, 23 30 40 45 50, se crea el nodo 4, se sube a raíz 40 como padre de los nodos 0 y 4.

L2, L0, E0, E4, E2

2: 0(40)4 (52)3 (70)1

0: (23)(30) 4:(45)(50) 3:(59)(65) 1: (80)(120)

+31 -> Se agrega sin problemas

L2, L0, E0

2: 0(40)4 (52)3 (70)1

0: (23)(30)(31) 4:(45)(50) 3:(59)(65) 1: (80)(120)

+34 -> Se agrega sin problemas

L2, L0, E0

2: 0(40)4 (52)3 (70)1

0: (23)(30)(31)(34) 4:(45)(50) 3:(59)(65) 1: (80)(120)

+38 -> Se produce overflow en el nodo 0, 23 30 31 34 38, se crea el nodo 5 y sube a raíz 31 como padre de los nodos 0 y 5.

L2, L0, E0, E5, E2

2: 0(31)5 (40)4 (52)3 (70)1

0: (23)(30) 5:(34)(38) 4:(45)(50) 3:(59)(65) 1: (80)(120)

+60 -> Se agrega sin problemas

L2, L3, E3

2: 0(31)5 (40)4 (52)3 (70)1

0: (23)(30) 5:(34)(38) 4:(45)(50) 3:(59)(60)(65) 1: (80)(120)

+63 -> Se agrega sin problemas

L2, L3, E3

2: 0(31)5 (40)4 (52)3 (70)1

0: (23)(30) 5:(34)(38) 4:(45)(50) 3:(59)(60)(63)(65) 1: (80)(120)

+64 -> Se produce overflow en el nodo 3, 59 60 63 64 65, se crea el nodo 6 y sube a raíz 63 como padre de los nodos 0 y 5. Se produce overflow en el nodo 2, 31 40 52 63 70, se crea el nuevo nodo 7, sube a raíz (nuevo nodo 8) como padre de 2 y 7 la clave 52.

L2, L3, E3, E6, E2, E7, E8

8: (52)

2: 0(31)5 (40)4 7: 3(63)6 (70)1

0: (23)(30) 5:(34)(38) 4:(45)(50) 3:(59)(60) 6:(64)(65) 1: (80)(120)

-23 -> Se elimina sin problemas

L8, L2, L0, E0

8: (52)

2: 0(31)5 (40)4 7: 3(63)6 (70)1

0: (30) 5:(34)(38) 4:(45)(50) 3:(59)(60) 6:(64)(65) 1: (80)(120)

-30 -> Se produce underflow, se trata de redistribuir con el adyacente izquierdo y como no hay lo hace con el derecho, 31 34 38, sube 34 como padre de los nodos 0 y 5.

L8, L2, L0, L5, E0, E5, E2.

8: (52)

2: 0(34)5 (40)4 7: 3(63)6 (70)1

0: (31) 5:(38) 4:(45)(50) 3:(59)(60) 6:(64)(65) 1: (80)(120)

-31 -> Se produce underflow, se trata de redistribuir con el adyacente izquierdo y como no hay lo hace con el derecho, como no tiene suficiente se produce una unión, se libera 5 y 40 queda como padre del nodo 0.

nodo libre: 5

L8, L2, L0, E0

8: (52)

2: 0(40)4 7: 3(63)6 (70)1

0: (34)(38) 4:(45)(50) 3:(59)(60) 6:(64)(65) 1: (80)(120)

-40 -> Se busca el menor del subárbol derecho suyo para reemplazarlo y se hace sin inconvenientes.

nodo libre: 5

L8, L2, L4, E2

8: (52)

2: 0(45)4 7: 3(63)6 (70)1

0: (34)(38) 4:(50) 3:(59)(60) 6:(64)(65) 1: (80)(120)

-45 -> Se busca el menor del subárbol derecho suyo para reemplazarlo, se produce underflow en el nodo 4, se busca en el adyacente izquierdo y se redistribuye, 34 38 50, 38 queda como raíz y padre de los nodos 0 y 4.

nodo libre: 5

L8, L2, L4, E2, L0, E0, E4, E2

8: (52)

2: 0(38)4 7: 3(63)6 (70)1

0: (34) 4:(50) 3:(59)(60) 6:(64)(65) 1: (80)(120)

-38 -> Se busca el menor del subárbol derecho suyo para reemplazarlo, se produce underflow en el nodo 4 y intenta redistribuir con el izquierdo, como no puede se produce una unión, quedando 34 50 en el nodo 0, liberando el 4. Esto produce un underflow en el nodo 2, aplica la política y como no hay adyacente izquierdo lo hace con el derecho, se produce una unión (esto se hace porque con una redistribución el nodo sigue quedando desbalanceado) en el nodo liberando el 7 y el 8

nodo libre: 8 7 5

L8, L2, L4, E2, L0, E0, L7, E2

2: 0(52) 3(63)6 (70)1

0: (34)(50) 3:(59)(60) 6:(64)(65) 1: (80)(120)

15- Árbol B con orden 6, ante disparidad de carga el nodo izquierdo más cargado y con resolución de underflow de izquierda o derecha.

2: 0 (216) 1 (460) 4 (570) 5 (689) 3 (777) 6

0: (100)(159)(171) 5: (586)(599)(615)(623)(680)

1: (222)(256)(358)(380)(423) 3: (703)(725)

4: (505)(522) 6: (789)(915)(1000)

+300 -> Se produce overflow en 1, 222 256 300 358 380 423, se crea nuevo nodo 7, sube 358 a raiz como padre de 1 y 7. Produce overflow, 216 358 460 570 689 777, crea nuevo nodo 8, sube 570 a raiz (en el nodo nuevo 9) como padre de 2 y 8.

L2, L1, E1, E7, E2, E8, E9

9: 2(570)8

2: 0 (216) 1(358)7 (460) 4 8: 5 (689) 3 (777) 6

0: (100)(159)(171) 5: (586)(599)(615)(623)(680)

1: (222)(256)(300) 3: (703)(725)

7:(380)(423) 6: (789)(915)(1000)

4: (505)(522)

+577 -> Produce overflow en el nodo 5, 577 586 599 615 623 680, se crea nuevo nodo 10 y 615 pasa al nodo 8 como padre del nodo 5 y 10

L9, L8, L5, E5, E10, E8

9: 2(570)8

2: 0 (216) 1(358)7 (460) 4 8: 5(615)10 (689) 3 (777) 6

0: (100)(159)(171) 5: (577)(586)(599)

1: (222)(256)(300) 10: (623)(680)

7:(380)(423) 3: (703)(725)

4: (505)(522) 6: (789)(915)(1000)

-586 -> Se elimina sin problemas

L9, L8, L5, E5

9: 2(570)8

2: 0 (216) 1(358)7 (460) 4 8: 5(615)10 (689) 3 (777) 6

0: (100)(159)(171) 5: (577)(599)

1: (222)(256)(300) 10: (623)(680)

7:(380)(423) 3: (703)(725)

4: (505)(522) 6: (789)(915)(1000)

-570 -> Produce underflow en nodo 5, aplica la política de underflow como no hay adyacente a la izquierda va a la derecha, no se puede redistribuir por lo que se hace una unión. Libero el nodo 10.

Nodo libre: 10

L9, L8, L5, E5, E8

9: 2(577)8

2: 0 (216) 1(358)7 (460) 4 8: 5 (689) 3 (777) 6

0: (100)(159)(171) 5: (599)(615)(623)(680)

1: (222)(256)(300)

7:(380)(423) 3: (703)(725)

4: (505)(522) 6: (789)(915)(1000)

-380 -> Produce underflow en nodo 7, aplica la política de underflow y redistribuye con el de la izquierda, 222 256 300 358 423, sube al nodo 2 la clave 300 como padre de los nodos 1 y 7

Nodos libres: 10

L9, L2, L7, L1, E1, E7, E2

9: 2(577)8

2: 0 (216) 1(300)7 (460) 4 8: 5(680)10 (689) 3 (777) 6

0: (100)(159)(171) 5: (599)(615)(623)(680)

1: (222)(256)

7:(358)(423) 3: (703)(725)

4: (505)(522) 6: (789)(915)(1000)

-460 -> Se reemplaza por su menor clave en el subárbol derecho, produce underflow en el nodo 4, se une con su nodo adyacente izquierdo (el 7), (358 423 505 522), 358 queda como padre del nodo 7. Queda libre el nodo 4

Nodos libres: 4 10

L9, L2, L4, E2, L7, E7, E2

9: 2(577)8

2: 0 (216) 1(300)7 (358) 4 8: 5(680)10 (689) 3 (777) 6

0: (100)(159)(171) 5: (599)(615)(623)(680)

1: (222)(256) 3: (703)(725)

7:(423)(505)(522) 6: (789)(915)(1000)

17- Árbol B+ de orden 4, ante disparidad de carga el nodo derecho debe quedar más cargado, con política de underflow a derecha.

2: 0 (66) 1

0: (23)(45) 1 1: (66) (67) (89)

+120 -> Overflow en nodo 1, 66 67 89 120

L2, L1, E1, E3, E2

2: 0 (66) 1 (67)3

0: (23)(45) 1 1: (66)(67) 3:(89)(120)

+110 -> Se agrega sin inconvenientes

L2, L3, E3

2: 0 (66) 1 (67)3

0: (23)(45) 1 1: (66)(67) 3:(89)(110)(120)

+52 -> Se agrega sin inconvenientes

L2, L0, E0

2: 0 (66) 1 (67)3

0: (23)(45)(52) 1 1: (66)(67) 3:(89)(110)(120)

+70 -> Se produce overflow en nodo 3, 70 89 110 120

L2, L1, E1, E4, E2

2: 0 (66) 1 (67)3 (89)4

0: (23)(45)(52) 1 1: (66)(67) 3:(70)(89) 4:(110)(120)

+15 -> Se produce overflow en nodo 0, 15 23 45 52, overflow en 2, 23 66 67 89

L2, L0, E0, E5, E2, E6, E7

7: 2(66)6

2: 0 (23)5 (66) 6: 1(67)3 (89)4

0: (15)(23) 5:(45) (52) 1 1: (67) 3:(70)(89) 4:(110)(120)

-45 -> Se elimina sin inconvenientes

L7, L2, L0, E0

7: 2(66)6

2: 0 (23)5 (66) 6: 1(67)3 (89)4

0: (15)(23) 5: (52) 1 1: (67) 3:(70)(89) 4:(110)(120)

-52 -> Se produce overflow en el nodo 5,

L7, L2, L5, E0, E2, E5

7: 2(66)6

2: 0 (15)5 (66) 6: 1(67)3 (89)4

0: (15) 5: (23) 1: (67) 3:(70)(89) 4:(110)(120)

+22 -> Se ingresa sin problemas

L7, L2, L5, E5

7: 2(66)6

2: 0 (15)5 (66) 6: 1(67)3 (89)4

0: (15) 5: (22)(23) 1: (67) 3:(70)(89) 4:(110)(120)

+19 -> Se ingresa sin problemas

L7, L2, L5, E5

7: 2(66)6

2: 0 (15)5 (66) 6: 1(67)3 (89)4

0: (15) 5: (19)(22)(23) 1: (67) 3:(70)(89) 4:(110)(120)

-66 -> Se elimina sin inconvenientes

L7, L2, L5, E5, E2

7: 2(66)6

2: 0 (15)5 (19) 6: 1(67)3 (89)4

0: (15) 5: (22)(23) 1: (67) 3:(70)(89) 4:(110)(120)

-22 -> Se elimina 19 sin inconvenientes.

L7, L2, L5, E5?, E2

7: 2(66)6

2: 0 (15)5 (22) 6: 1(67)3 (89)4

0: (15) 5: (23) 1: (67) 3:(70)(89) 4:(110)(120)

-19 -> Underflow en nodo 5, 15 23 se unen, libera el 2, underflow en nodo 2, se unen el 7 y el 6, libera el 6

nodos libres: 6,2

L7, L2, L5, E0, L6, E2

2: 2(66)6 1(67)3 (89)4

0: (15) (23) 1: (67) 3:(70)(89) 4:(110)(120)

-23 -> Se elimina sin inconvenientes.

nodos libres: 6,2

L2, L0, E0

2: 2(66)6 1(67)3 (89)4

0: (15) 1: (67) 3:(70)(89) 4:(110)(120)

-89 -> Se elimina sin inconvenientes.

nodos libres: 6,2

L2, L3, E3

2: 2(66)6 1(67)3 (89)4

0: (15) 1: (67) 3:(70) 4:(110)(120)

18-

19- Dado el siguiente árbol B+ de orden 4, mostrar cómo quedaría definido el árbol luego de realizar las siguientes operaciones: -56, -23, -1.

2: 0 (56) 1

0: (1)(23) 1 1: (56) (75) (107)

-56 -> Se elimina sin problemas

L2, L1, E1

2: 0 (56) 1

0: (1)(23) 1 1: (75) (107)

-23 -> Se elimina sin problemas

L2, L0, E0

2: 0 (56) 1

0: (1) 1 1: (75) (107)

-1 -> Se produce underflow, 75 107, 107 queda como raíz y padre de los nodos 0 y 1.

L2, L0, L1, E0

2: 0(107)1

0: (75) 1 1: (107)

20- Dado el siguiente árbol B+ de orden 4 , mostrar cómo quedaría el árbol luego de realizar cada operación : +96, -10, -67, -23, -37.

2: 0 (37) 3 (67) 1

0: (10)(23) 3 3: (37)(45) 1 1: (67) (77)

+96 -> Se agrega sin problemas

L2, L1, E1

2: 0 (37) 3 (67) 1

0: (10)(23) 3 3: (37)(45) 1 1: (67) (77) (96)

-10 -> Se elimina sin problemas

L2, L0, E0

2: 0 (37) 3 (67) 1

0: (23) 3 3: (37)(45) 1 1: (67) (77) (96)

-67 -> Se elimina sin problemas

L2, L1, E1

2: 0 (37) 3 (67) 1

0: (23) 3 3: (37)(45) 1 1: (77) (96)

-23 -> Se produce underflow, 37 45, se crea una copia de 45 que será el padre de los nodos 0 y 3

L2, L0, E0, E3, E2

2: 0 (45) 3 (67) 1

0: (37) 3 3: (45) 1 1: (77) (96)

-37 -> Se produce underflow, cuando intenta redistribuir su nodo derecho no tiene suficiente por lo que se produce una unión y se libera el nodo derecho, quedando todo en el nodo 0.

L2, L0, E0, E2

2: 0 (67) 1

0: (45) 1 1: (77) (96)

21- Dado el árbol B+ que se detalla más abajo, con orden 6, es decir, capacidad de 5 claves como máximo. Muestre los estados sucesivos al realizar la siguiente secuencia de operaciones: +159, -5 y -190, además indicar nodos leídos y escritos en el orden de ocurrencia. Política de resolución underflow derecha.

Nodo 2: 5, i, 0(10)1(60)3(115)4(145)5(179)6 Nodo

0: 2, h, (1)(5) -> 1 Nodo

1: 2, h, (34)(44) -> 3 Nodo

3: 2, h, (60)(113) -> 4 Nodo

4: 4, h, (120)(125)(131)(139) -> 5 Nodo

5: 5, h, (145)(153)(158)(160)(177) -> 6 Nodo

6: 2, h, (179)(190) -> -1

+159 -> Se produce overflow en el nodo 5, 145 153 158 159 160 177, se crea nuevo nodo 7, sube el 159 al nodo 2, se produce overflow en 2, 10 60 115 145 159 179, se crea nuevo nodo 8, 145 sube al nodo 9 que sera la raiz como padre de 8 y 2

L2, L5, E5, E7, E2, E8, E9

9: 1, i, 2(145)8

2: 3, i, 0(10)1(60)3(115)4 8: 3, i, (145)5(159)7(179)6 -> -1

0: 2, h, (1)(5) -> 1 5: 5, h, (145)(153)(158) -> 7

1: 2, h, (34)(44) -> 3 7: 3, h, (159)(160)(177) -> 6

3: 2, h, (60)(113) -> 4 6: 2, h, (179)(190) -> -1

4: 4, h, (120)(125)(131)(139) -> 5

-5 -> Se produce underflow en el nodo 0, como no puede redistribuir se unen el nodo 0 y 1, se libera el nodo 1, 1 34 44

L9, L2, L0, L1, E0, E2

9: 1, i, 2(145)8

2: 3, i, 0(44)1(60)3(115)4 8: 3, i, (145)5(159)7(179)6 -> -1

0: 2, h, (1)(34) -> 1 5: 5, h, (145)(153)(158) -> 7

1: Libre 7: 3, h, (159)(160)(177) -> 6

3: 2, h, (60)(113) -> 4 6: 2, h, (179)(190) -> -1

4: 4, h, (120)(125)(131)(139) -> 5

-190 -> Se produce underflow en el nodo 6, se redistribuye con el nodo 7, 159 160 177 179, sube 177 como padre de los nodos 7 y 6 al nodo 8.

L9, L2, L0, L1, E0

9: 1, i, 2(145)8

2: 3, i, 0(34)1(60)3(115)4 8: 3, i, (145)5(159)7(177)6 -> -1

0: 2, h, (1)(44) -> 1 5: 5, h, (145)(153)(158) -> 7

1: Libre 7: 3, h, (159)(160) -> 6

3: 2, h, (60)(113) -> 4 6: 2, h, (177)(179) -> -1

4: 4, h, (120)(125)(131)(139) -> 5

22- Dado un árbol B de orden 5 y con política izquierda o derecha.

8: 1 i 2 (150) 7

2: 1 i 0 (120) 3 7: 2 i 4 (210)6(300)1

0: 2 h (30)(110) 4: 4 h (160)(170)(180)(200)

3: 1 h (130) 6: 4 h (220)(230)(240)(250)

1: 4 h (400)(500)(600)(700)

+165 -> Overflow en el nodo 4, 160 165 170 180 200, se crea nuevo nodo 9, sube 170 a nodo 7 como padre de los nodos 4 y 9.

L8, L7, L4, E4, E9, E7

8: 1 i 2 (150) 7

2: 1 i 0 (120) 3 7: 2 i 4(170)9 (210)6(300)1

0: 2 h (30)(110) 4: 3 h (160)(165)

3: 1 h (130) 9: 3 h (180)(200)

6: 4 h (220)(230)(240)(250)

1: 4 h (400)(500)(600)(700)

+260 -> Overflow en el nodo 6, 220 230 240 250 260, se crea nuevo nodo 10, sube 240 a nodo 7 como padre de los nodos 6 y 10.

L8, L7, L6, E6, E10, E7

8: 1 i 2 (150) 7

2: 1 i 0 (120) 3 7: 2 i 4(170)9 (210)6 (240)10 (300)1

0: 2 h (30)(110) 4: 2 h (160)(165)

3: 1 h (130) 9: 2 h (180)(200)

6: 2 h (220)(230)

10: 2 h (250)(260)

1: 4 h (400)(500)(600)(700)

+800 -> Overflow en el nodo 1, 400 500 600 700 800, se crea nuevo nodo 11, sube 600 a nodo 7 como padre de los nodos 1 y 11. Se produce overflow, 170 210 240 300 600, se crea nuevo nodo 12, sube 240 como padre de 7 y 12

L8, L7, L1, E1, E11, E7, E12, E8

8: 1 i 2 (150) 7 (240) 12

2: 1 i 0 (120) 3 7: 2 i 4(170)9 (210)6 12: 2 i 10 (300)1 (600)11

0: 2 h (30)() 4: 2 h (160)(165) 10: 2 h (250)(260)

3: 1 h (130) 9: 2 h (180)(200) 1: 4 h (400)(500)

6: 2 h (220)(230) 11: 2h (700)(800)

-110 -> Se elimina sin inconvenientes

L8, L2, L0, E0

8: 1 i 2 (150) 7 (240) 12

2: 1 i 0 (120) 3 7: 2 i 4(170)9 (210)6 12: 2 i 10 (300)1 (600)11

0: 2 h (30) 4: 2 h (160)(165) 10: 2 h (250)(260)

3: 1 h (130) 9: 2 h (180)(200) 1: 4 h (400)(500)

6: 2 h (220)(230) 11: 2h (700)(800)

23- Dado un árbol B+ de orden 5 y con política izquierda o derecha.

nodo 8: 1 i 2(70)7

2:1 i 0(50)4 7: 4 i 5(90)6(120)3(210)9(300)1

0: 1 h(40)->4 5: 2 h(70)(80)->6

4: 1 h(50)->5 6: 2 h(90)(100)->3

3: 2 h(120)(200)->9

9: 4 h(210)(220)(230)(240)->1

1: 2 h(300)(400)(500)-> -1

+240 -> El nodo ya existe

L8, L7, L9

nodo 8: 1 i 2(70)7

2:1 i 0(50)4 7: 4 i 5(90)6(120)3(210)9(300)1

0: 1 h(40)->4 5: 2 h(70)(80)->6

4: 1 h(50)->5 6: 2 h(90)(100)->3

3: 2 h(120)(200)->9

9: 4 h(210)(220)(230)(240)->1

1: 2 h(300) (400)(500)-> -1

-300 -> El nodo se elimina sin inconvenientes

L8, L7, L1,E1

nodo 8: 1 i 2(70)7

2:1 i 0(50)4 7: 4 i 5(90)6(120)3(210)9(300)1

0: 1 h(40)->4 5: 2 h(70)(80)->6

4: 1 h(50)->5 6: 2 h(90)(100)->3

3: 2 h(120)(200)->9

9: 4 h(210)(220)(230)(240)->1

1: 2 h(400)(500)-> -1

-40 -> Se produce underflow en el nodo 0, se busca en el nodo adyacente de su izquierda para redistribuir y no existe, lo mismo con el derecho pero no tiene suficientes, hace con el nodo 4. Nos queda el nodo 0 con 50 y se libera el nodo 4. Se produce underflow en el nodo 2 y como el nodo está desbalanceado se produce una redistribución, 70 90 120 210 300

L8, L2, L0, L4, E0, E2, E7, E8

nodo 8: 1 i 2(120)7

2:2 i 0(70)5 (90)6 7: 2 i 3(210)9(300)1

0: 1 h(50)->5 3: 2 h(120)(200)->9

5: 2 h(70)(80)->6 9: 4 h(210)(220)(230)(240)->1

6: 2 h(90)(100)->3 1: 2 h(400)(500)-> -1