

Introducción a las Bases de Datos

Fundamentos de Organización de Datos

Práctica 4

Políticas para la resolución de underflow:

Política izquierda: se intenta redistribuir con el hermano adyacente izquierdo, si no es posible, se fusiona con hermano adyacente izquierdo.

Política derecha: se intenta redistribuir con el hermano adyacente derecho, si no es posible, se fusiona con hermano adyacente derecho.

Política izquierda o derecha: se intenta redistribuir con el hermano adyacente izquierdo, si no es posible, se intenta con el hermano adyacente derecho, si tampoco es posible, se fusiona con hermano adyacente izquierdo.

Política derecha o izquierda: se intenta redistribuir con el hermano adyacente derecho, si no es posible, se intenta con el hermano adyacente izquierdo, si tampoco es posible, se fusiona con hermano adyacente derecho.

Casos especiales: en cualquier política si se tratase de un nodo hoja de un extremo del árbol debe intentarse redistribuir con el hermano adyacente que el mismo posea.

Aclaración:

- En caso de underflow lo primero que se intenta **SIEMPRE** es redistribuir y el hermano adyacente se encuentra en condiciones de ceder un elemento si al hacerlo no se produce underflow en el.

Árboles B y B+

1. Definir la estructura de datos correspondiente a un árbol B de orden M, que almacenará información correspondiente a los médicos de un centro privado. De los mismos deberá guardarse nombre y apellido, dni, matricula y año de ingreso. ¿Cuál de estos datos debería seleccionarse como clave de identificación para organizar los elementos en el árbol? ¿Hay más de una opción? Justifique su elección.
2. Redefinir la estructura de datos del ejercicio anterior para un árbol B+ de orden M. Responda detalladamente:

- a. ¿Cómo accede a la información para buscar al médico con dni 37.222.111?
- b. ¿Cómo accede a la información para buscar al médico García Mariano?
- c. Indique cuáles son las ventajas que ofrece este tipo de árbol para el caso de la búsqueda planteada en el inciso b.

3. Suponga ahora que los datos de los médicos se encuentran desordenados en un archivo de datos y Ud. debe proveer mecanismos para la búsqueda de médicos por los siguientes criterios: dni, legajo y apellido y nombre. Defina las estructuras de datos necesarias para estos requerimientos.

4. Dado el siguiente algoritmo de búsqueda en un árbol B:

```
function buscar(NRR, clave, NRR_encontrado, pos_encontrada)
begin
  if (nodo = nulo)
    buscar := false; {clave no encontrada}
  else
    posicionarYLeerNodo(A, nodo, NRR);
    if (claveEncontrada(A, nodo, clave, pos)) then
      NRR_encontrado := NRR; {NRR actual}
      pos_encontrada := pos; {posición dentro del array}
    end
  else
    buscar(nodo.hijo[pos], clave, NRR_encontrado, pos_encontrada)
  end;
end;
```

Asuma que para la primera llamada, el parámetro NRR contiene la posición de la raíz del árbol. Responda detalladamente:

- a. PosicionarYLeerNodo(): Indique qué hace y la forma en que deben ser enviados los parámetros (valor o referencia).
- b. claveEncontrada(): Indique qué hace y la forma en que deben ser enviados los parámetros (valor o referencia). ¿Cómo lo implementaría?
- c. ¿Existe algún error en este código? En caso afirmativo, modifique lo que considere necesario.

5. Defina los siguientes conceptos:

- Overflow
- Underflow
- Redistribución
- Fusión o concatenación

En los dos últimos casos, ¿cuándo se aplica cada uno?

Para los siguientes ejercicios debe:

- **Indicar los nodos leídos y escritos en cada operación.**
- **Todas las operaciones deben estar claramente justificadas, enunciando las mismas indefectiblemente tal cual se presenta en la materia.**
- **Los números de nodo deben asignarse en forma coherente con el crecimiento del archivo. La reutilización de nodos libres se debe efectuar con política LIFO (último en entrar, primero en salir).**

6. Dado el siguiente árbol B de orden 5, mostrar como quedaría el mismo luego de realizar las siguientes operaciones: +445, -490, -507, -608. Justificar detalladamente cada operación indicando lecturas y escrituras en orden de ocurrencia. Para la resolución de underflow debe utilizar **política a izquierda**. Graficar cada operación por separado.

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)

7. Suponga que usted es administrador de un servidor de correo electrónico. Los logs del mismo (información guardada acerca de los movimientos que ocurren en el server) se encuentran almacenados en el archivo logmail.dat. De los mismos se guarda la siguiente información: número de usuario, nombre de usuario, nombre, apellido y cantidad de mails enviados. Definir la estructura de datos correspondiente a un árbol B de orden M, que almacenará la información correspondiente al archivo logmail.dat. Además, declarar la variable correspondiente al árbol que se va a generar.

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

Política de resolución de underflows: derecha.

Política de resolución de underflows: derecha o izquierda.

Política de resolución de underflows: 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)

12. Dado el siguiente árbol B de orden 5 , mostrar como quedaría el mismo luego de realizar las siguientes operaciones: +15, -45, -50.

Ante disparidad de carga, el nodo izquierdo debe quedar más cargado.

Política de resolución de underflows: izquierda.

2: 0 (56) 1

0:
(10)(35)(45)(50)

1: (66) (70) (88)(120)

13. Dada las siguientes operaciones, mostrar la construcción paso a paso de un árbol B de orden 4: +50 , +70, +40, +15, +90, +120, +115, +45, +30, +100, +112, +77, -45, -40, -50, -90, -100.

Ante disparidad de carga, el nodo izquierdo debe quedar más cargado.

Política de resolución de underflows: izquierda o derecha.

14. Dadas las siguientes operaciones, mostrar la construcción paso a paso de un árbol B de orden 5:

Ante disparidad de carga, el nodo izquierdo debe quedar más cargado.

Política de resolución de underflows: izquierda.

+80, +50, +70, +120, +23, +52, +59, +65, +30, +40, +45, +31, +34, +38, +60, +63, +64, -23, -30, -31, -40, -45, -38.

15. Dado el siguiente árbol B de orden 6, mostrar como quedaría el mismo luego de realizar las siguientes operaciones: +300, +577, -586, -570, -380, -460.

Ante disparidad de carga, el nodo izquierdo debe quedar más cargado.

Política de resolución de underflows: 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)

16. Dada las siguientes operaciones, mostrar cómo se construye el árbol B de orden 4: +65, +89, +23, +45, +20, +96, +10, +55, -23, +110, +50, -10, +25, -50, -45, +120, +130, +70, +75, +73, +100, -120, -110.

Ante disparidad de carga, el nodo derecho debe quedar más cargado.

Política de resolución de underflows: derecha.

17. Dado el siguiente árbol B+ de orden 4, mostrar como quedaría el mismo luego de realizar las siguientes operaciones: +120, +110, +52, +70, +15, -45, -52, +22, +19, -66, -22, -19, -23, -89.

Ante disparidad de carga, el nodo derecho debe quedar más cargado.

Política de resolución de underflows: derecha.

2: 0 (66) 1

0: (23)(45) 1

1: (66) (67) (89)

18. Dada las siguientes operaciones, mostrar la construcción paso a paso de un árbol B+ de orden 4:

+67, +56, +96, +10, +28, +95, +16, +46, +23, +36, +120, +130, +60, +57, -96, -67, -95, -60, -120, -57, -56.

Ante disparidad de carga, el nodo derecho debe quedar más cargado.

Política de resolución de underflows: derecha o izquierda.

Dada las siguientes operaciones, mostrar la construcción paso a paso de un árbol B+ de orden 6:

+52, +23, +10, +99, +63, +74, +19, +85, +14, +73, +5, +7, +41, +100, +130, +44, -63, -73, +15, +16, -74, -52.

Ante disparidad de carga, el nodo derecho debe quedar más cargado.

Política de resolución de underflows: izquierda.

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)

20. Dado el siguiente árbol B+ de orden 4 , mostrar como 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)