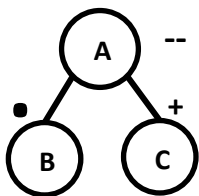




CP 5: AVL

- 1) Responda Verdadero (V) o Falso (F) según corresponda. Justifique su respuesta en cada caso:
 - a. ☒ En un árbol AVL puede existir un nodo con altura 15 con un hijo de altura 12
 - b. ☒ El número máximo de rotaciones que se pueden hacer en el algoritmo de eliminación en el árbol AVL es 3. (Considere la rotación doble como 2 rotaciones)
 - c. ☒ En un AVL, después de una eliminación en la que se efectuaron rotaciones, si existen nodos no hojas, al menos uno de ellos tiene factor de balance 0.
 - d. ☒ En todo AVL el valor absoluto de la diferencia entre las longitudes de dos ramas cualesquiera es a lo sumo 1.

- 2) Para la situación de desbalance en un AVL mostrada en la figura (siguiendo el convenio visto en conferencia para el factor de balance), responda Verdadero o Falso y justifique su respuesta:

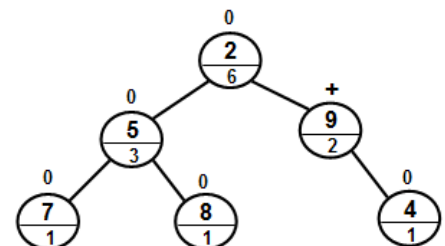


- a. ☒ Luego de resolverla los factores de balance quedarán A-, B+, C+.
- b. ☒ Esta situación puede aparecer en la inserción y en la eliminación.

- ☒ 3) Sea A un árbol AVL de n nodos que almacena números reales. Implemente un algoritmo que determine en $O(\log n + k)$ la cantidad k de elementos de A que se encuentran en un intervalo dado $[a, b]$. Justifique el tiempo de complejidad y la correctitud de su algoritmo.
 - ☒ a. ¿Qué modificación podría hacerse a la estructura de este árbol para que se puedan obtener la cantidad de elementos de A que se encuentran en un intervalo dado $[a, b]$ en $O(\log n)$?
 - ☒ b. Implemente un algoritmo que devuelva en $O(\log n)$ la cantidad elementos de A que están en el intervalo $[a, b]$, dados a y b.

- 4) En un árbol AVL se le ha agregado a cada nodo el campo *Peso*. Para cada nodo, dicho campo indica la cantidad de nodos que hay en el subárbol donde él es raíz (incluyéndolo a él). Con esta modificación al AVL y, eliminando además el criterio de los menores por la izquierda y los mayores por la derecha, se puede hacer uso de él para implementar una lista de manera eficiente. Por ejemplo: la figura muestra un AVL con las consideraciones anteriores y donde la lista que él representa se corresponde con el recorrido en simétrico.

- ☒ a. Plantee el algoritmo para la operación *Buscar*(AVL, i) que devuelve en $O(\log n)$ el i -ésimo elemento en el recorrido en simétrico (elemento en la posición i de la lista).
- ☒ b. Plantee un algoritmo $O(\log n)$ para la operación *Insertar*(AVL, i, x) que incluye a x en el árbol de manera tal que: el recorrido en simétrico del árbol resultante, es el mismo que la lista que se obtiene al insertar a x , en la posición i de la lista representada por el árbol inicial. ¿Qué operación (u operaciones) es necesario modificar para garantizar que el árbol resultante tenga correctamente actualizado el campo *Peso* y el *Factor de Balance*? Diga en cada caso cuál es la modificación





5) Se desea Implementar la operación *JOIN* que permita la unión de dos AVL:

- ✓ a. Implemente la operación *JOIN*(A, x, B) de donde sabemos que todo valor del árbol AVL A es menor que el valor del elemento x, y a su vez, el elemento x es menor que todo valor del árbol AVL B. Esta operación devuelve un árbol AVL con los valores de A, los valores de B y el valor x, en $O(\max\{\text{altura}(A), \text{altura}(B)\})$.
- ? b. Implemente la operación *JOIN*(A, B) de donde sabemos que todo valor del árbol AVL A es menor que todo valor del árbol AVL B. Esta operación devuelve un árbol AVL con los valores de A y los valores de B, en $O(\max\{\text{altura}(A), \text{altura}(B)\})$.

? 6) Implemente la operación *SPLIT*(A, x) que devuelve dos AVL donde en el primero estarán todos los valores de A menores o iguales que x y en el segundo los mayores que este. Su tiempo de complejidad será $O(\log^2 n)$ donde n es la cantidad de nodos de A. Justifique el tiempo de complejidad.

- 7) Plantee una modificación al árbol AVL que permita implementar la operación *JOIN*(A, x, B) en orden $O(|\text{altura}(A) - \text{altura}(B)|)$ y la operación *SPLIT*(A, x) en orden $O(\log(n))$ donde n es la cantidad de nodos de A.
- a. ¿Qué modificaciones hay que hacer a las operaciones en el AVL, producto de la modificación, de forma talque el orden de complejidad de cada una de ellas no varíe? Justifique su respuesta.
 - b. Describa en pseudocódigo las operaciones *JOIN* y *SPLIT*. Justifique el orden de complejidad en cada una de ellas.

8) Para el siguiente método que recibe un arreglo de enteros y construye un árbol binario a partir del mismo:

```
class Nodo {
    public Nodo HijoIzq {get .. set ..}
    public Nodo HijoDerecho {get .. set ..}
    public int Valor {get .. set ..}
    public int Factor_Balance { get .. set ..}
}
Node Construye (int[] A) {
    return Construye (A, 0, A. Length - 1);
}
Node Construye(int[] A, int i, int j) {
    int m = (i + j) / 2;
    Nodo node = new Node {Valor = A[m]};
    // línea en blanco
    if (i < m) node.HijoIzquierdo = Construye(A, i, m - 1);
    if (m < j) node.HijoDerecho = Construye(A, m + 1, j);
    return node;
}
```

- ✓ a. Demuestre que el árbol resultante es un árbol AVL. (Hint: Demuestre que todos los niveles, excepto quizás el último, están completos)
- ✓ b. Agregue al algoritmo, en la línea en blanco, instrucciones, con costo total $O(1)$, para asignar el factor de balance al nodo *node*. Demuestre el funcionamiento de las mismas.