

PARTE 2 - Equipos: Ejercicios de pseudocódigo

Duración: 60 minutos

EJERCICIO 1

Los árboles binarios de búsqueda son usados para organizar un conjunto de elementos con el objetivo de que las búsquedas sobre él tengan un orden del tiempo de ejecución logarítmico en el mejor de los casos. Este mejor caso corresponde a que el árbol resulte armado completo, con todas sus hojas al mismo nivel.

Para asegurar un orden logarítmico en las búsquedas en todos los casos, se introduce el criterio de balance "AVL" que establece que, para cada nodo del árbol, la diferencia de alturas de sus dos subárboles no puede ser mayor a uno.

Insertar las siguientes claves en el orden dado en un árbol AVL:

74, 25, 10, 61, 35, 47, 18, 20, 21, 23, 40, 55, 63.

CONSIDERACIONES IMPORTANTES

- Se debe mostrar paso a paso la inserción de cada clave.
- Cuando se produce un desbalance, se debe identificar el nodo desbalanceado, el tipo de desbalance y los distintos nodos que se ven involucrados en la operación de balanceo (k1, k2, k3).
- En caso de rotaciones complejas, se puede obviar mostrar el estado intermedio luego de realizar la primera rotación.

EJERCICIO 2

El sistema de inventario representa un almacén que contiene una variedad de productos. Cada producto está asociado con un código único y se organiza jerárquicamente en categorías y subcategorías. Los nodos internos del árbol representan estas categorías, mientras que los nodos hoja representan los productos individuales sin subcategorías.

Dado un árbol de búsqueda binaria (ABB) que representa un sistema de gestión de inventario, se solicita implementar un algoritmo para recorrer el árbol y separar sus nodos en dos listas: nodos hoja (productos sin más subcategorías) y nodos internos (categorías que contienen subcategorías/productos). Se debe mantener el orden relativo de los nodos dentro de cada lista.

Las firmas de todos los métodos son abiertas, pero se debe incluir una demostración de cómo invocar al método solicitado.

CONSIDERACIONES IMPORTANTES

- Deben desarrollarse en detalle todos los métodos que se invoquen.
- Se tendrá en cuenta la eficiencia de los algoritmos desarrollados, calificando mejor aquellos que tengan mejor tiempo de ejecución o mejor performance general.
- No es necesario desarrollar los métodos de los TDAs utilizados si estos corresponden a las variantes vistas en clase.
- Se deben cumplir todos los pasos estándar de desarrollo de algoritmos, Lenguaje Natural, Pre y Post Condiciones, Pseudocódigo y Análisis del Orden del Tiempo de Ejecución