

Ejercicios de árboles binarios de búsqueda

1. Implementa un método que reciba un nodo y que devuelva su recorrido por niveles.
2. En la clase de árbol binario de búsqueda, implementa un método recursivo que recorra el árbol e imprima por pantalla los elementos cuyo abuelo tengan un valor múltiplo de 10.
3. Desarrolla un método recursivo que sume todos los elementos del árbol y que devuelva la suma total.
4. Desarrolla un método (iterativo) que dado un nodo binario de búsqueda devuelva su nodo predecesor.
5. Desarrolla un método (iterativo) que dado un nodo binario de búsqueda devuelva su nodo sucesor.
6. En la clase árbol binario de búsqueda, desarrolla un método que busque y devuelva el elemento con la clave más pequeña.
7. En la clase árbol binario de búsqueda, desarrolla un método que busque y devuelva el elemento con la clave de mayor valor.
8. Crea un método que calcule el factor de equilibrio por tamaño de un determinado nodo. El factor de equilibrio por tamaño de un nodo es igual al valor absoluto de la diferencia entre el tamaño de su hijo derecho y su hijo izquierdo.
9. Crea un método que calcule el factor de equilibrio por altura de un determinado nodo. El factor de equilibrio por altura de un nodo es igual al valor absoluto de la diferencia entre la altura de su hijo derecho y de su hijo izquierdo.
10. En la clase de árbol binario de búsqueda, añade un método que compruebe si el árbol está perfectamente equilibrado (todos sus nodos tienen factor de equilibrio en tamaño menor o igual que 1).
11. En la clase de árbol binario de búsqueda, añade un método que compruebe si el árbol está equilibrado en altura (todos sus nodos tienen factor de equilibrio en altura menor o igual que 1).
12. Crea una clase Test y en su método main, crea un árbol binario de búsqueda de claves y elementos Integer e inserta la siguiente secuencia de elementos (k,e):
(70,70),(20,20),(15,15),(10,10),(2,2),(59,59),(29,29),(69,69),(98,98),
(77,77),(75,75),(91,91),(78,78),(86,86)

