

## GUÍA SOBRE BSTs

1. Hacer una función en C++ que, dado un BST T y un entero N, borre todos los nodos cuyo dato sea igual a N.
2. Hacer una función en C++ que, dado un árbol T, determine si es un BST válido.
3. Hacer una función en C++ que, dado un arreglo de enteros, los traslade a un BST.
4. Modificar el ejercicio anterior, para que el BST resultante sea lo más balanceado posible.
5. Hacer una función en C++ que, dado un arreglo de enteros que representa la secuencia IN ORDER de un BST, construya dicho BST.
6. Hacer una función en C++ que, dado un arreglo de enteros que representa la secuencia PRE ORDER de un BST, construya dicho BST.
7. Hacer una función en C++ que, dado un arreglo de enteros que representa la secuencia POST ORDER de un BST, construya dicho BST.
8. Hacer una función en C++ que, dado un BST y un dato, encuentre el sucesor IN ORDER del nodo que contiene dicho dato.
9. Hacer una función en C++ que, dado un BST y un dato, encuentre el sucesor PRE ORDER del nodo que contiene dicho dato.
10. Hacer una función en C++ que, dado un BST y un dato, encuentre el sucesor POST ORDER del nodo que contiene dicho dato.
11. Hacer una función en C++ que, dado un BST y un dato, encuentre el antecesor IN ORDER del nodo que contiene dicho dato.
12. Hacer una función en C++ que, dado un BST y un dato, encuentre el antecesor PRE ORDER del nodo que contiene dicho dato.
13. Hacer una función en C++ que, dado un BST y un dato, encuentre el antecesor POST ORDER del nodo que contiene dicho dato.
14. Hacer una función en C++ que, dado un BST y dos enteros A y B, borre del BST todos los nodos que contengan datos en el rango [A, B].
15. Hacer una función en C++ que, dado un BST y un entero N, encuentre dos nodos en el BST cuyos datos sumen N.
16. Hacer una función en C++ que, dado un BST de enteros, transforme el contenido de cada nodo a su correspondiente Código ASCII.
17. Hacer una función en C++ que haga el proceso opuesto del ejercicio anterior.
18. Hacer una función en C++, que dada una lista enlazada y ordenada de enteros L, coloque todos los datos en un BST.
19. Hacer una función en C++ que, dado un BST, lo transforme en un MIN-HEAP.
20. Hacer una función en C++ que haga el proceso opuesto del ejercicio anterior.
21. Hacer una función en C++ que, dado un BST, lo transforme en un MAX-HEAP.
22. Hacer una función en C++ que haga el proceso opuesto del ejercicio anterior.
23. Hacer una función en C++ que, dados dos BST, los fusione en uno solo.
24. Hacer una función en C++ que, dado un BST, verifique si todos los nodos tienen exactamente 1 hijo.
25. Hacer una función en C++ que, dados 2 BST, verifique si son iguales.
26. Hacer una función en C++ que, dado un BST y un entero N, encuentre el dato más grande en el BST que sea menor o igual a N.
27. Hacer una función en C++ que, dado un BST y un entero N, encuentre el dato más pequeño en el BST que sea mayor o igual a N.

28. Hacer una función en C++ que, dado un BST y dos datos, encuentre la distancia entre los nodos que contengan los dos datos dados. (La distancia entre dos nodos es la cantidad de nodos en el camino en el árbol desde el primero hasta el último).
29. Hacer una función en C++ que, dado un BST, borre todas las hojas.
30. Hacer una función en C++ que, dado un BST y un entero N, encuentre el nodo con el dato más cercano a N.
31. Hacer una función en C++ que, dado un BST, verifique que esté completo. (Un árbol es completo, si en el penúltimo nivel todos los nodos tienen 2 hijos).
32. Hacer una función en C++ que, dado un BST y un entero N, devuelva la profundidad del nodo que contenga el dato N. (La profundidad es la distancia desde la raíz hasta el nodo).
33. Hacer una función en C++ que, dado un BST y dos enteros A y B, verifique si A es ancestro de B en el BST.
34. Hacer una función en C++ que, dado un BST y un entero N, devuelva una lista enlazada circular con todos los datos menores o iguales a N que estén en el BST.
35. Hacer una función en C++ que, dada una lista doblemente enlazada, cree un BST con los datos de la lista recorriéndola de 2 en 2.