<https://www.interviewcake.com/concept/java/tree>

* [Check if two given binary trees are identical or not](https://www.techiedelight.com/check-if-two-binary-trees-are-identical-not-iterative-recursive/)
* [Calculate height of a binary tree](https://www.techiedelight.com/calculate-height-binary-tree-iterative-recursive/)
* [Delete given Binary Tree](https://www.techiedelight.com/delete-given-binary-tree-iterative-recursive/)
* [Inorder Tree Traversal (Iterative & Recursive Implementation)](https://www.techiedelight.com/inorder-tree-traversal-iterative-recursive/)
* [Preorder Tree Traversal (Iterative & Recursive Implementation)](https://www.techiedelight.com/preorder-tree-traversal-iterative-recursive/)
* [Postorder Tree Traversal (Iterative & Recursive Implementation)](https://www.techiedelight.com/postorder-tree-traversal-iterative-recursive/)
* [Level Order Traversal of Binary Tree](https://www.techiedelight.com/level-order-traversal-binary-tree/)
* [Spiral Order Traversal of Binary Tree](https://www.techiedelight.com/spiral-order-traversal-binary-tree/) (practice)
* [Reverse Level Order Traversal of Binary Tree](https://www.techiedelight.com/reverse-level-order-traversal-binary-tree/)
* [Print left view of binary tree](https://www.techiedelight.com/print-left-view-of-binary-tree/) (Important)
* [Print Bottom View of Binary Tree](https://www.techiedelight.com/print-bottom-view-of-binary-tree/) (Important) (unsolved)
* [Print Top View of Binary Tree](https://www.techiedelight.com/print-top-view-binary-tree/) (Important)
* [Find next node in same level for given node in a binary tree](https://www.techiedelight.com/find-next-node-in-same-level-binary-tree/)
* [Check if given binary tree is complete binary tree or not](https://www.techiedelight.com/check-given-binary-tree-complete-binary-tree-not/)
* [In-place convert given binary tree to its sum tree](https://www.techiedelight.com/inplace-convert-a-tree-sum-tree/)
* [Determine if given two nodes are cousins of each other](https://www.techiedelight.com/determine-two-nodes-are-cousins/)
* [Print cousins of given node in a binary tree](https://www.techiedelight.com/print-cousins-of-given-node-binary-tree/)
* [Check if given binary tree is a sum tree or not](https://www.techiedelight.com/check-given-binary-tree-sum-tree-not/)
* [Combinations of words formed by replacing given numbers with corresponding alphabets](https://www.techiedelight.com/combinations-of-words-formed-replacing-given-numbers-corresponding-english-alphabet/)
* [Determine if given binary tree is a subtree of another binary tree or not](https://www.techiedelight.com/determine-given-binary-tree-is-subtree-of-another-binary-tree-not/)
* [Find diameter of a binary tree](https://www.techiedelight.com/find-diameter-of-a-binary-tree/)
* [Check if given binary Tree has symmetric structure or not](https://www.techiedelight.com/check-given-binary-tree-symmetric-structure-not/)
* [Convert binary tree to its mirror](https://www.techiedelight.com/convert-binary-tree-to-its-mirror/)
* [Check if binary tree can be converted to another by doing any no. of swaps of left & right child](https://www.techiedelight.com/determine-binary-tree-can-converted-another-number-swaps-left-right-child/)
* [Find Lowest Common Ancestor (LCA) of two nodes in a binary tree](https://www.techiedelight.com/find-lowest-common-ancestor-lca-two-nodes-binary-tree/)
* [Print all paths from root to leaf nodes in a binary tree](https://www.techiedelight.com/print-all-paths-from-root-to-leaf-nodes-binary-tree/)
* [Find ancestors of given node in a Binary Tree](https://www.techiedelight.com/find-ancestors-of-given-node-binary-tree/)
* [Find the distance between given pairs of nodes in a binary tree](https://www.techiedelight.com/distance-between-given-pairs-of-nodes-binary-tree/)
* [Find Vertical Sum in a given Binary Tree](https://www.techiedelight.com/find-vertical-sum-given-binary-tree/)
* [Perform vertical traversal of a binary tree — I](https://www.techiedelight.com/vertical-traversal-binary-tree/)
* [Perform vertical traversal of a binary tree — II](https://www.techiedelight.com/print-nodes-binary-tree-vertical-order/)
* [Print corner nodes of every level in binary tree](https://www.techiedelight.com/print-corner-nodes-every-level-binary-tree/)
* [Find the diagonal sum of given binary tree](https://www.techiedelight.com/find-diagonal-sum-given-binary-tree/)
* [Print Diagonal Traversal of Binary Tree](https://www.techiedelight.com/print-diagonal-traversal-binary-tree/)
* [In-place convert Binary Tree to Doubly Linked List](https://www.techiedelight.com/place-convert-given-binary-tree-to-doubly-linked-list/)
* [Sink nodes containing zero to the bottom of the binary tree](https://www.techiedelight.com/sink-nodes-containing-zero-bottom-binary-tree/)
* [Convert given binary tree to full tree by removing half nodes](https://www.techiedelight.com/convert-given-binary-tree-to-full-tree-removing-half-nodes/)
* [Truncate given binary tree to remove nodes which lie on a path having sum less than K](https://www.techiedelight.com/truncate-given-binary-tree-remove-nodes-lie-path-sum-less-k/)
* [Find maximum sum root-to-leaf path in a binary tree](https://www.techiedelight.com/find-maximum-sum-root-to-leaf-path-binary-tree/)
* [Check if given binary tree is height balanced or not](https://www.techiedelight.com/check-given-binary-tree-is-height-balanced-not/)
* [Find maximum width of given binary tree](https://www.techiedelight.com/find-maximum-width-given-binary-tree/)
* [Convert normal binary tree to Left-child right-sibling binary tree](https://www.techiedelight.com/convert-normal-binary-tree-left-child-right-sibling-binary-tree/)
* [Determine if given Binary Tree is a BST or not](https://www.techiedelight.com/determine-given-binary-tree-is-a-bst-or-not/)
* [Convert a Binary Tree to BST by maintaining its original structure](https://www.techiedelight.com/convert-binary-tree-to-bst-maintaining-original-structure/)
* [Invert a Binary Tree](https://www.techiedelight.com/invert-binary-tree-recursive-iterative/)
* [Print Right View of a Binary Tree](https://www.techiedelight.com/print-right-view-binary-tree/)
* [Print all paths from leaf to root node in given binary tree](https://www.techiedelight.com/print-all-paths-from-leaf-to-root-binary-tree/)
* [Iteratively print leaf to root path for every leaf node in a binary tree](https://www.techiedelight.com/print-leaf-to-root-path-binary-tree/)
* [Build Binary Tree from given Parent array](https://www.techiedelight.com/build-binary-tree-given-parent-array/)
* [Find all nodes at given distance from leaf nodes in a binary tree](https://www.techiedelight.com/find-all-nodes-at-given-distance-from-leaf-nodes-in-a-binary-tree/)
* [Count all subtrees having same value of nodes in a binary tree](https://www.techiedelight.com/count-subtrees-value-nodes-binary-tree/)
* [Find Maximum Difference Between a Node and its Descendants in a Binary Tree](https://www.techiedelight.com/find-maximum-difference-node-descendants/)
* [Construct a Binary Tree from Ancestor Matrix](https://www.techiedelight.com/construct-binary-tree-ancestor-matrix/)
* [Calculate height of a binary tree with leaf nodes forming a circular doubly linked list](https://www.techiedelight.com/calculate-height-binary-tree-leaf-nodes-forming-circular-doubly-linked-list/)
* [Find maximum sum path between two leaves in a binary tree](https://www.techiedelight.com/find-maximum-sum-path-between-two-leaves-in-a-binary-tree/)
* [Fix a binary tree that is only one swap away from becoming a BST](https://www.techiedelight.com/fix-binary-tree-one-swap-bst/)
* [Construct a binary tree from inorder and preorder traversal](https://www.techiedelight.com/construct-binary-tree-from-inorder-preorder-traversal/)
* [Construct a binary tree from inorder and postorder traversals](https://www.techiedelight.com/construct-binary-tree-from-inorder-postorder-traversals/)
* [Construct a binary tree from inorder and level order sequence](https://www.techiedelight.com/construct-binary-tree-from-inorder-level-order-traversals/)
* [Construct a full binary tree from preorder sequence with leaf node information](https://www.techiedelight.com/construct-full-binary-tree-from-preorder-sequence-with-leaf-information/)
* [Construct a full binary tree from a preorder and postorder sequence](https://www.techiedelight.com/construct-full-binary-tree-from-preorder-postorder-sequence/)
* [Set next pointer to inorder successor of all nodes in binary tree](https://www.techiedelight.com/set-next-pointer-inorder-successor-binary-tree/)
* [Efficiently print all nodes between two given levels in a binary tree](https://www.techiedelight.com/print-nodes-between-two-levels-binary-tree/)
* [Find preorder traversal of a binary tree from its inorder and postorder sequence](https://www.techiedelight.com/find-preorder-traversal-binary-tree-from-inorder-postorder/)
* [Find the difference between sum of all nodes present at odd and even levels in a binary tree](https://www.techiedelight.com/difference-between-sum-nodes-odd-even-levels/)
* [Find the size of the largest BST in a Binary Tree](https://www.techiedelight.com/find-size-largest-bst-in-binary-tree/)
* [Link nodes present in each level of a binary tree in the form of a linked list](https://www.techiedelight.com/link-nodes-each-level-binary-tree/)
* [Construct a Cartesian Tree from In-order Traversal](https://www.techiedelight.com/construct-cartesian-tree-from-inorder-traversal/)
* [Implementation of Treap Data Structure (Insert, Search and Delete)](https://www.techiedelight.com/implementation-treap-data-structure-cpp-java-insert-search-delete/)
* [Clone a binary tree with random pointers](https://www.techiedelight.com/clone-a-binary-tree-with-random-pointers/)
* [Threaded Binary Tree: Overview and Implementation](https://www.techiedelight.com/threaded-binary-tree-overview-implementation/)
* [Invert alternate levels of a perfect binary tree](https://www.techiedelight.com/invert-alternate-levels-perfect-binary-tree/)
* [Convert a Binary Tree into a Doubly Linked List in Spiral Order](https://www.techiedelight.com/convert-binary-tree-into-doubly-linked-list/)
* [Check if a binary tree is a min-heap or not](https://www.techiedelight.com/check-binary-tree-is-min-heap/)
* [Determine if a binary tree satisfy the height-balanced property of red–black tree](https://www.techiedelight.com/determine-binary-tree-satisfy-height-balanced-property-red-black-tree/)
* [Depth first search (DFS) vs Breadth first search (BFS)](https://www.techiedelight.com/depth-first-search-dfs-vs-breadth-first-search-bfs/)

<https://www.interviewbit.com/courses/programming/topics/tree-data-structure/>

**List of problems sorted in increasing order of difficulty:**

1. [**Pre-order Traversal of a Binary Tree**](https://www.ideserve.co.in/learn/preorder-traversal-of-a-binary-tree)
2. [**Post-order Traversal of a Binary Tree**](https://www.ideserve.co.in/learn/postorder-traversal-of-a-binary-tree)
3. [**In-order Traversal of a Binary Tree**](https://www.ideserve.co.in/learn/inorder-traversal-of-a-binary-tree)
4. [**Binary Tree Level Order Traversal**](https://www.ideserve.co.in/learn/binary-tree-level-order-traversal)
5. [**Print right view of a binary tree**](https://www.ideserve.co.in/learn/right-view-of-a-binary-tree)
6. [**Print all nodes of a binary tree that do not have sibling**](https://www.ideserve.co.in/learn/print-nodes-of-binary-tree-without-sibling)
7. [**Print all Root to Leaf paths of a Binary Tree**](https://www.ideserve.co.in/learn/print-all-root-to-leaf-paths-of-a-binary-tree)
8. [**Minimum Depth of a Binary Tree**](https://www.ideserve.co.in/learn/minimum-depth-of-a-binary-tree)
9. [**Print left view of a binary tree**](https://www.ideserve.co.in/learn/left-view-of-a-binary-tree)
10. [**Find sum of all left leaves of a binary tree**](https://www.ideserve.co.in/learn/find-sum-of-all-left-leaves-binary-tree)
11. [**Find depth of deepest odd level leaf node**](https://www.ideserve.co.in/learn/depth-of-deepest-odd-level-leaf-node)
12. [**Check whether a binary tree is a full binary tree or not**](https://www.ideserve.co.in/learn/check-whether-binary-tree-is-full-binary-tree-or-not)
13. [**Check whether a binary tree is complete or not**](https://www.ideserve.co.in/learn/check-whether-binary-tree-is-complete-tree-or-not)
14. [**Check if two nodes are cousins in a Binary tree**](https://www.ideserve.co.in/learn/check-if-two-nodes-are-cousins-binary-tree)
15. [**Check if two binary trees are identical**](https://www.ideserve.co.in/learn/check-if-two-binary-trees-are-identical)
16. [**Check if all internal nodes of BST have only one child without building tree**](https://www.ideserve.co.in/learn/check-if-all-internal-nodes-have-one-child-bst)
17. [**Convert the given n-ary tree to its mirror image**](https://www.ideserve.co.in/learn/mirror-of-n-ary-tree)
18. [**Convert a binary tree to its mirror tree**](https://www.ideserve.co.in/learn/mirror-a-tree)
19. [**Print top view of a binary tree**](https://www.ideserve.co.in/learn/top-view-of-a-binary-tree)
20. [**Print top view of a binary tree using level order traversal**](https://www.ideserve.co.in/learn/top-view-of-a-binary-tree-using-level-order-traversal)
21. [**Print bottom view of a binary tree**](https://www.ideserve.co.in/learn/bottom-view-of-a-binary-tree)
22. [**Print bottom view of a binary tree using level order traversal**](https://www.ideserve.co.in/learn/bottom-view-of-a-binary-tree-using-level-order-traversal)
23. [**Remove the nodes of binary search tree which are outside the given range**](https://www.ideserve.co.in/learn/remove-out-of-range-bst-nodes)
24. [**Remove all nodes which lie on path having sum less than k**](https://www.ideserve.co.in/learn/remove-all-nodes-which-lie-on-path-having-sum-less-than-k)
25. [**Remove all the half nodes from a given binary tree**](https://www.ideserve.co.in/learn/remove-all-half-nodes-binary-tree)
26. [**Print binary tree in vertical order**](https://www.ideserve.co.in/learn/print-binary-tree-vertical-order)
27. [**Populate right neighbors for all nodes in a binary tree**](https://www.ideserve.co.in/learn/populate-right-neighbors-in-a-binary-tree)
28. [**Lowest Common Ancestor of two nodes in a Binary Search Tree**](https://www.ideserve.co.in/learn/lowest-common-ancestor-of-two-nodes-binary-search-tree)
29. [**Iterative Pre-order Traversal of a Binary Tree**](https://www.ideserve.co.in/learn/iterative-preorder-traversal-of-binary-tree)
30. [**In-order Successor of a Node in a Binary Tree**](https://www.ideserve.co.in/learn/inorder-successor-of-a-node-in-a-binary-tree)
31. [**Recover a Binary Search Tree if positions of two nodes are swapped.**](https://www.ideserve.co.in/learn/how-to-recover-a-binary-search-tree-if-two-nodes-are-swapped)
32. [**Find floor and ceiling of an element from given dataset using binary search tree**](https://www.ideserve.co.in/learn/floor-ceiling-using-binary-search-tree)
33. [**Diagonal Sum of a Binary Tree.**](https://www.ideserve.co.in/learn/diagonal-sum-of-a-binary-tree)
34. [**Create a balanced Binary Search Tree from a sorted array**](https://www.ideserve.co.in/learn/create-a-balanced-bst-from-a-sorted-array)
35. [**Convert a sorted Doubly Linked List to Balanced Binary Search Tree**](https://www.ideserve.co.in/learn/convert-a-sorted-doubly-linked-list-to-balanced-binary-search-tree-bst)
36. [**Convert a binary tree to doubly linked list**](https://www.ideserve.co.in/learn/convert-a-binary-tree-to-doubly-linked-list)
37. [**Check if a binary tree is balanced or not**](https://www.ideserve.co.in/learn/check-if-a-binary-tree-is-balanced)
38. [**Check if a binary tree is a binary search tree**](https://www.ideserve.co.in/learn/check-if-a-binary-tree-is-a-binary-search-tree)
39. [**Check if two binary search trees are identical given their array representations | Set 2**](https://www.ideserve.co.in/learn/check-if-identical-binary-search-trees-without-building-them-set-2)
40. [**Check if two binary search trees are identical given their array representations**](https://www.ideserve.co.in/learn/check-if-identical-binary-search-trees-without-building-them-set-1)
41. [**Check if a binary tree is sub-tree of another binary tree in time O(n)**](https://www.ideserve.co.in/learn/check-if-a-binary-tree-is-subtree-of-another-binary-tree-time-optimized)
42. [**Check if a binary tree is sub-tree of another binary tree in space O(1)**](https://www.ideserve.co.in/learn/check-if-a-binary-tree-is-subtree-of-another-binary-tree-space-optimized)
43. [**Binary Search tree | Insertion and Search**](https://www.ideserve.co.in/learn/binary-search-tree-insertion)
44. [**Binary Search tree | Deletion**](https://www.ideserve.co.in/learn/binary-search-tree-delete)
45. [**Check if a given binary tree is symmetric tree or not**](https://www.ideserve.co.in/learn/check-if-binary-tree-is-symmetric-tree)
46. [**Check if the given n-ary tree is symmetric tree or not**](https://www.ideserve.co.in/learn/check-if-n-ary-tree-is-symmetric-tree)
47. [**Spiral Level Order Traversal of a Binary Tree | Set 1**](https://www.ideserve.co.in/learn/spiral-level-order-traversal-of-a-binary-tree-set-1)
48. [**Spiral Level Order Traversal of a Binary Tree | Set 2**](https://www.ideserve.co.in/learn/spiral-level-order-traversal-of-a-binary-tree-set-2)
49. [**Find maximum element from each sub-array of size 'k'| Set 1**](https://www.ideserve.co.in/learn/maximum-element-from-each-subarray-of-size-k-set-1)
50. [**Total number of possible Binary Search Trees with 'n' keys**](https://www.ideserve.co.in/learn/total-number-of-possible-binary-search-trees-with-n-keys)
51. [**Find the size of largest BST in a binary tree**](https://www.ideserve.co.in/learn/size-of-largest-bst-in-binary-tree)
52. [**Lowest Common Ancestor of 2 nodes in a Binary Tree**](https://www.ideserve.co.in/learn/lowest-common-ancestor-binary-tree)
53. [**Find height of the binary tree from its parent array representation**](https://www.ideserve.co.in/learn/find-height-of-binary-tree-from-parent-array)
54. [**Convert binary tree to binary search tree**](https://www.ideserve.co.in/learn/convert-binary-tree-to-binary-search-tree)
55. [**Construct the binary tree from its parent array representation**](https://www.ideserve.co.in/learn/construct-binary-tree-from-parent-array)
56. [**Construct binary tree from inorder and preorder traversals**](https://www.ideserve.co.in/learn/construct-binary-tree-from-inorder-and-preorder-traversals)
57. [**Construct binary tree from inorder and postorder traversals**](https://www.ideserve.co.in/learn/construct-binary-tree-from-inorder-and-postorder-traversals)
58. [**AVL tree | Basics**](https://www.ideserve.co.in/learn/avl-tree)
59. [**AVL tree | Insertion**](https://www.ideserve.co.in/learn/avl-tree-insertion)
60. [**AVL tree | Deletion**](https://www.ideserve.co.in/learn/avl-tree-deletion)
61. [**Trie Data Structure | Insert and search**](https://www.ideserve.co.in/learn/trie-insert-and-search)
62. [**Trie Data Structure | Delete**](https://www.ideserve.co.in/learn/trie-delete)
63. [**Pattern matching using Trie**](https://www.ideserve.co.in/learn/pattern-matching-using-trie)
64. [**Longest Prefix Matching using Trie**](https://www.ideserve.co.in/learn/longest-prefix-match-using-trie)
65. [**Given a sequence of words, group together all anagrams and print them.**](https://www.ideserve.co.in/learn/anagram-grouping-in-a-sequence-using-trie)
66. [**Serialize and Deserialize a binary search tree**](https://www.ideserve.co.in/learn/serialize-deserialize-binary-search-tree)
67. [**Serialize and Deserialize a binary search tree using post order traversal**](https://www.ideserve.co.in/learn/serialize-deserialize-binary-search-tree-using-post-order-traversal)