Topcoder archives - Trees and Bst – A set of hard conceptual problems by Alexander Mitrovic @slackcobra

1. Print nodes at k distance from root iter and rec ii. Print nodes at K distance from a given node iii. Print all nodes that are at a K distance from a leaf node 2. Count leaves iter and rec 3. Diagonal traversal 4. Iterative i. Pre 2. Post 3. In 5. Morris order traversal for i. In ii. Pre 6. Reverse Level order 7. Level order in spiral i. iter ii. Rec 8. Populate successors inorder 9. Successor and predecessor inorder 43. 10. Check if a bt is bst 11. Horizontal distance 12. Remove duplicates of a given key 13. Check for children sum property 14. Order statistics bst 15. Kth greatest node bst 16. Level order traversal line by line 17. Check if all levels of two trees are anagrams or not 18. Average of levels in a bt 19. Maximum sum from a tree with adjacent levels not allowed 20. Check if leaf traversals of two trees are same 21. Lca 22. Distance 23. Diameter 24. Print in a vertical order 25. Diagonal sum 26. Boundary traversal 27. Remove keys outside given range 28. Floor and ceil 29. Sorted linked list to balanced bst 30. Sorted array to bst 31. Check for identical bst without constructing the actual bsts from arrays 32. Find a pair with given sum in a balanced BST 33. Binary tree to bst conversion 34. Min value from a bst 35. Max value from a bst 36. Deletion from bst 37. Construction of bst i. lot ii. Pre iii. Post or bt from given traversal i. LOT and In ii. Post and In iii. Pre and In 38. Sorted order printing of a given array that represents a BST 39. Minimum Possible value of |ai + aj – k| for given array and k 40. Shortest path between two nodes in array like representation of binary tree 41. Remove nodes on root to leaf paths of length < K 42. Print common nodes on path from root (or common ancestors) 43. Replace each node in binary tree with the sum of its inorder predecessor and successor 44. Query for ancestor-descendant relationship in a tree 45. Two nodes of a BST are swapped, correct the BST 46. Print BST keys in the given range 47. Find the largest BST subtree in a given Binary Tree 48. Merge two balanced bst ii. Merge two bst with constant extra space 49. Generate all possible permutations in C 50. Generate all subsets in C 51. BST to heap conversion 52. Heap to BST conversion 53. Check if each internal node of a BST has exactly one child 54. Convert a BST to a Binary Tree such that sum of all greater keys is added to every key 55. Add all greater values to every node in a given BST 56. Find if there is a triplet in a Balanced BST that adds to zero 57. Count pairs from two BSTs whose sum is equal to a given value x 58. Largest number in BST which is less than or equal to N 59. Find median of BST in O(n) time and O(1) space 60. Check if an array represents Inorder of Binary Search tree or not 61. Transform a BST to greater sum tree 62. Given n appointments, find all conflicting appointments 63. In-place Convert BST into a Min-Heap 64. Check whether BST contains Dead End or not 65. Find the closest element in Binary Search Tree 66. Find pairs with given sum such that pair elements lie in different BSTs 67. Replace every element with the least greater element on its right 68. Check if given sorted sub-sequence exists in binary search tree 69. Print Common Nodes in Two Binary Search Trees 70. How to implement decrease key or change key in Binary Search Tree 71. Count BST nodes that lie in a given range 72. Count BST subtrees that lie in given range 73. Leaf nodes from Preorder of a Binary Search Tree 74. Maximum element between two nodes of BST 75. Binary Search Tree insert with Parent Pointer 76. BST to dll in place with changing pointers 77. BST of strings is given. Print all anagrams by combining one or more strings if possible. Print all such possible strings. 78. Convert a BST to fenwick tree(Binary Inedexed Tree) by changing the addresses of current pointers 79. Permute all nodes and convert the given BST to a permutation tree(Decision tree)

A special thanks to Rajaneesh @devrezo, Hailey @cynede_hl and Peter Rusev @prusev for your contributions to this catalogue.