1. What is the minimum and maximum number of keys in the 2-3-4 tree which has the depth 4 and which contains exactly one 4-node?

2. Suppose that a 2-3-4 tree is originally empty. Insert, in the given order, into the tree the keys

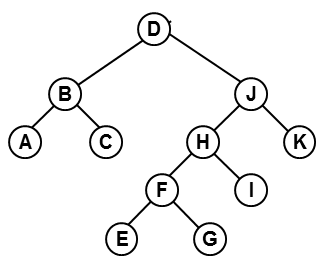
23, 31, 15, 24, 36, 20, 32, 18, 59, 60, 58, 57. Draw the tree after each insertion.

3. A splay tree is empty. Insert, in the given order, the keys 2, 7, 1, 4, 3, 9, 5, 6. Draw the tree after each insertion.

4. A Splay tree contains N keys,, two of them are labeled by x and y. A sequence of total 2N commands was issued: (Find(x), Find(y), Find(x), Find(y), ..., Find(x), Find(y)). What was the average number of nodes visited during one Find operation? Establish an upper and a lower bound of the average, as tight as possible.

5. A splay tree contains 2*n* −1 keys which values are 1, 2, 3, ..., 2*n* −1. The tree is ideally balanced, its depth is *n*−1.

Operation Find(1) splays the key 1 to the tree root. What is the depth of the resulting tree? Investigate odd and even values of *n* separately.



6. Assign red or black color A. B. C.

to each node so that the resulting tree

is a correct RB tree.

Note that the black [nil] leaves

are not depicted here.

7. Suppose that black height of a RB tree is 11. In this tree, determine the maximum number of

A) black nodes, B) red nodes, C) all nodes.

8. RB tree contains 15 keys and its black depth is 2,i.e. the tree contains 10 red nodes. The key values are

1, 2, 3, ..., 7, 8, 20, 21, 22, ..., 25, 26. Next, keys with values 10, 11, 12 are inserted one by one into this tree. Draw the original tree and the tree after each insertion.

9. Decide, if there exists a regular (each internal node has 2 children) binary search tree which cannot be turned to a RB tree by some (clever) coloring of its nodes. Find an example of such tree or argue that such tree cannot exist.

10. Two empty binary search trees are isomorphic. Let T1 and T2 be two unempty binary trees with the respective roots R1 and R2. T1 and T2 are isomorphic iff :

(R1.L is isomprphic to R2.L and R1.R is isomorphic to R2.R ) or

(R1.L is isomprphic to R2.R and R1.R is isomorphic to R2.L ).

The symbols ".L" and ".R" denote the left and the right subtree.

What is the number of non-isomorphic binary search trees with 2, 3, ..., 12 nodes?

Describe a recurrent relation which solves the problem in general case i.e. in case of N nodes in the tree.

11. Two empty 2-3-4 trees are isomorphic. Let T1 and T2 be two unempty 2-3-4 trees with the respective roots R1 and R2. T1 and T2 are isomorphic iff both 1. and 2. holds:

1. The root of T1 contains the same number of keys as the root of T1

2. The leftmost subtree of R1 is isomorphic to the leftmost subtree of R2, the second subtree of R1 from left is isomorphic to the second subtree of R2 from left, etc., and finaly the rightmost subtree of R1 is isomorphic to the rightmost subtree of R2.

What is the number of non-isomorphic 2-3-4 trees with 2, 3, ..., 12 nodes?

12. Define the relation of isomorphism between two R-B trees. Find the number of non-isomorphic R-B trees with 2, 3, ..., 12 nodes.

13. Suppose that a B+ tree of order 1 is originally empty. Insert, in the given order, into the tree the keys

32, 18, 31, 59, 20, 23, 24, 36, 60, 58, 15, 57, 51, 17, 16, 26, 42, 21, 43, 12. Draw the tree after each insertion.

Next, delete the keys from the tree in the order:

23, 31, 26, 15, 24, 42, 17, 36, 20, 43, 16, 32, 18, 59, 21, 51, 60, 12, 58, 57. Draw the tree after each insertion.

14. Write a pseudocode (or, indeed, a code) which will determine the number of non-isomorphic 2-3-4 trees with fixed depth D. Alternatively, you might derive a closed algebraic fomula which returns the number depending only on the value of D.