RB Trees

CS112 Recitation

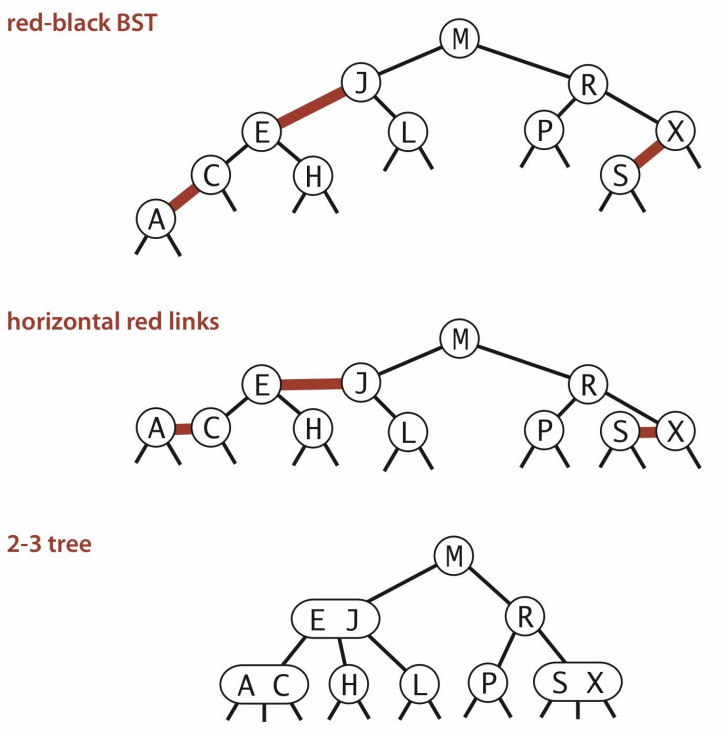
Original slides by Hoai-An Nguyen

Reminder: Exam 2 Thursday

1. Linked Structures
    a. SLL, DLL, CLL
        i. Review your exam 1, slides, and recitation problems
2. Decision Trees
    a. Binary Search Algorithm
        i. 2/25 recitation
        ii. Know how to find total # of comparisons for best case success, worst case success, failure, average, etc.
3. BST
    a. Be familiar with the three types of traversals, insertion, deletion, search, etc.
    b. Coding problems from recitation
4. Balanced Search Trees
    a. 2-3 Trees & Red-Black Trees
        i. Know how to search and insert in both types of trees
        ii. Understand the code for search/insert in Red-Black trees
5. Big Oh!

Let's Review: Red-Black Trees

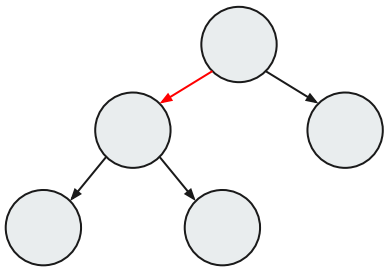
- Red-black BSTs are simple implementations of 2-3 trees
- RB BSTs have 3 restrictions:
  1. Red links lean left
  2. No node has two red links connected to it
  3. The tree has perfect black balance such that every path from the root to a null link has the same number of black links



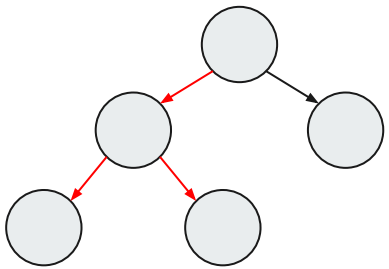
Warm-Up

1. True or false. If you insert keys in increasing order into a red-black BST, the tree height is monotonically increasing.
2. Which RB trees are invalid, and what rules do they break?

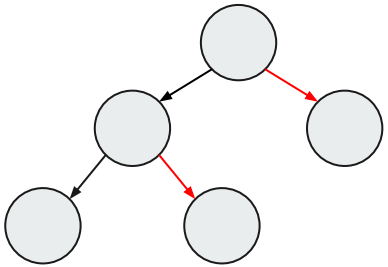
A



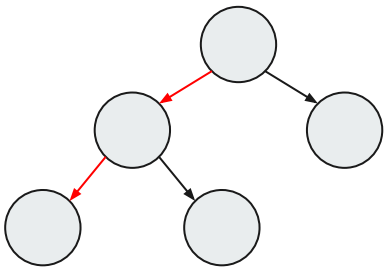
B



C

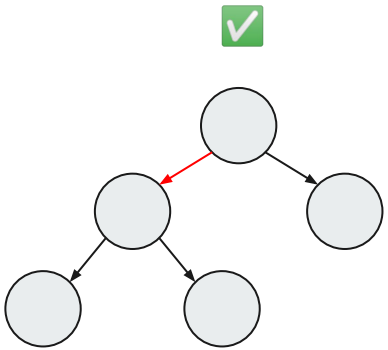


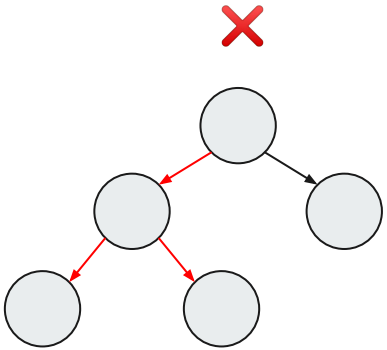
D



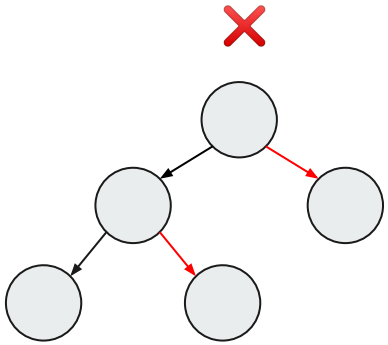
Warm-Up Answers

1. True or false. If you insert keys in increasing order into a red-black BST, the tree height is monotonically increasing. **True!**
2. Which RB trees are invalid, and what rules do they break?

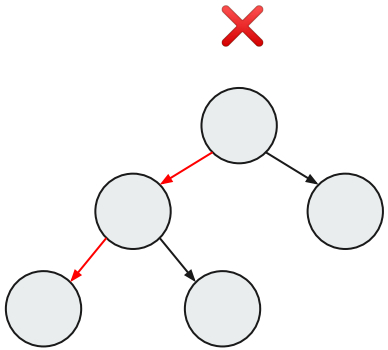




No node can have two red links connected to it



Red links must lean left

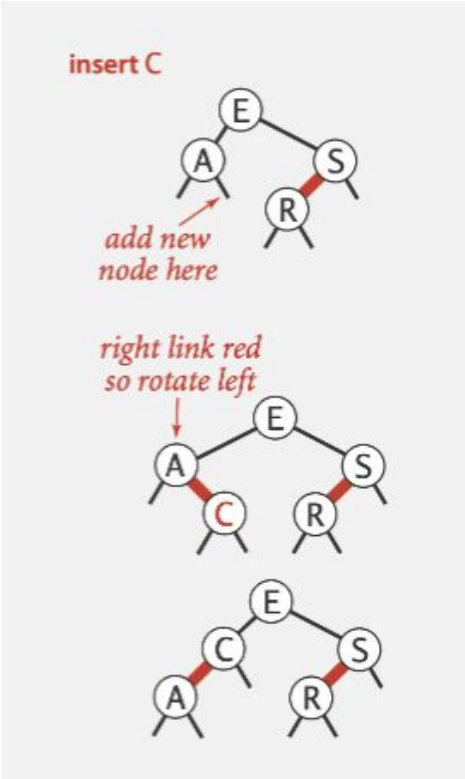


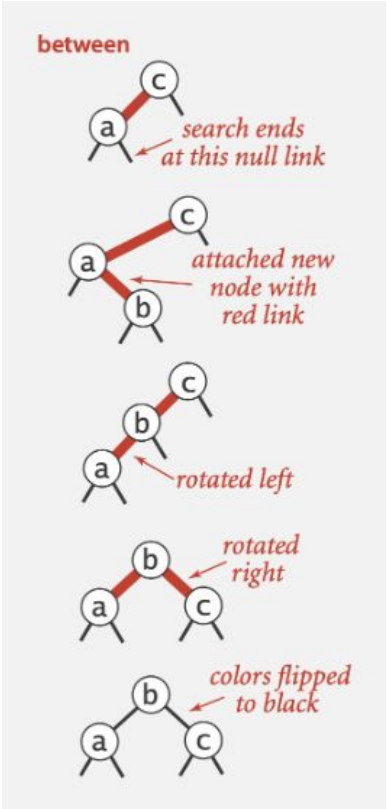
Tree does not have perfect black balance (count the black links!)

How to Insert?

1. Do standard BST insert, color link red
2. Check for violations:
    a. Right-leaning red link
        i. Rotate left
    b. Two red links in a row
        i. Rotate right
    c. Right link and left link are BOTH red (this is a 4-node in a 2-3 tree)
        i. Both links turn black, and the parent becomes red
        ii. (if the parent is the root, you can keep it black)
3. After every rotation, check for violations
4. If there are no violations, backtrack and check for violations at the parent, and so on...

How to Insert?



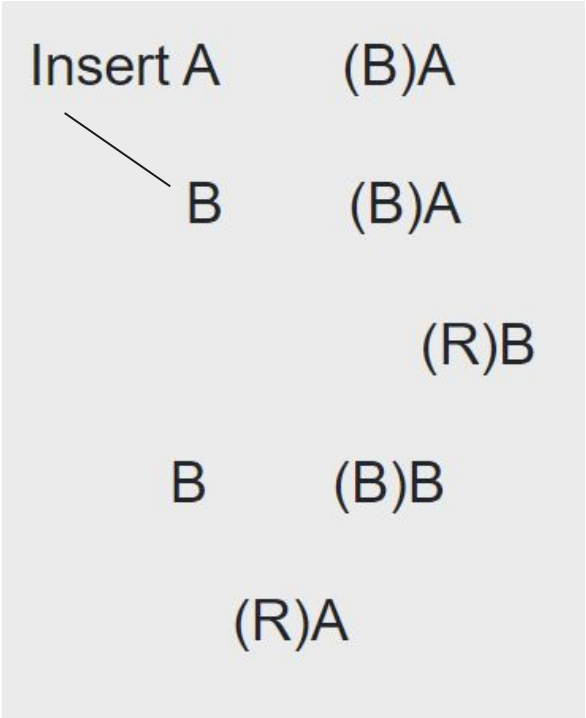


RB - Insert Ascending Order

Draw the red-black BST that results when you insert letters A through K in order into an initially empty tree, then describe what happens in general when trees are built by insertion of keys in ascending order.

Answer: Insert A, B

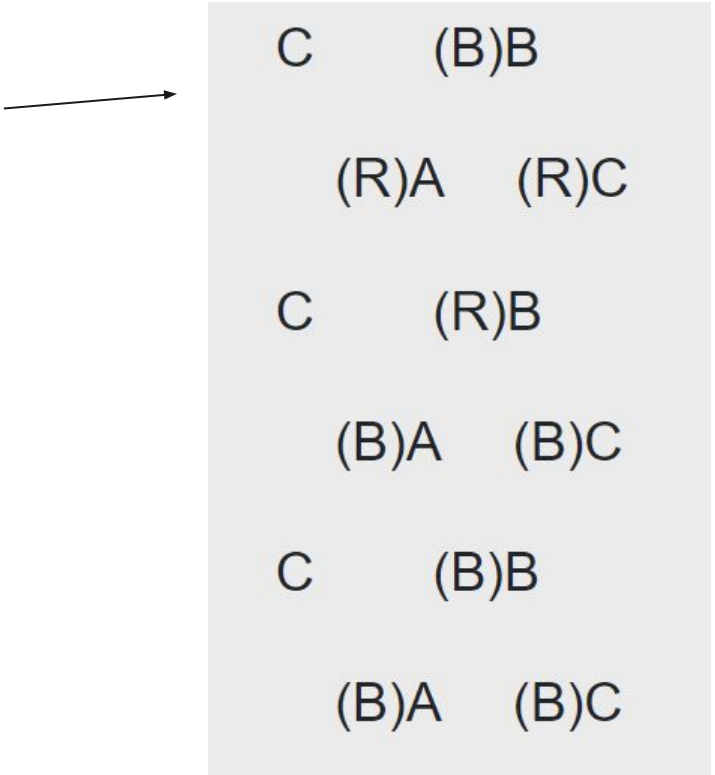
Answer: Insert A, B



Answer: Insert C

Answer: Insert C

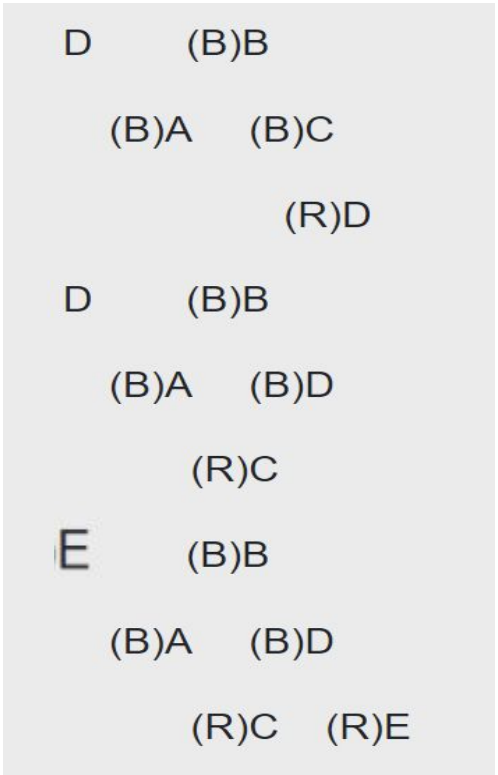
Insert

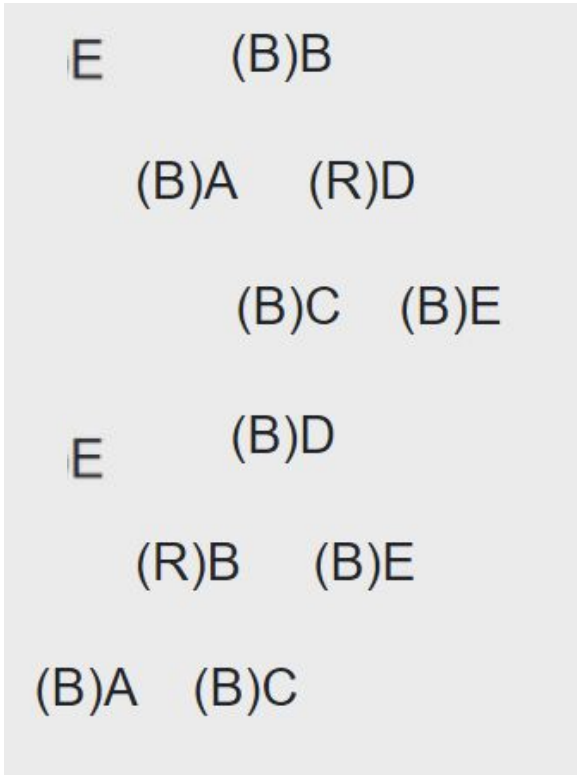


Answer: Insert D, E

Answer: Insert D, E

Insert →

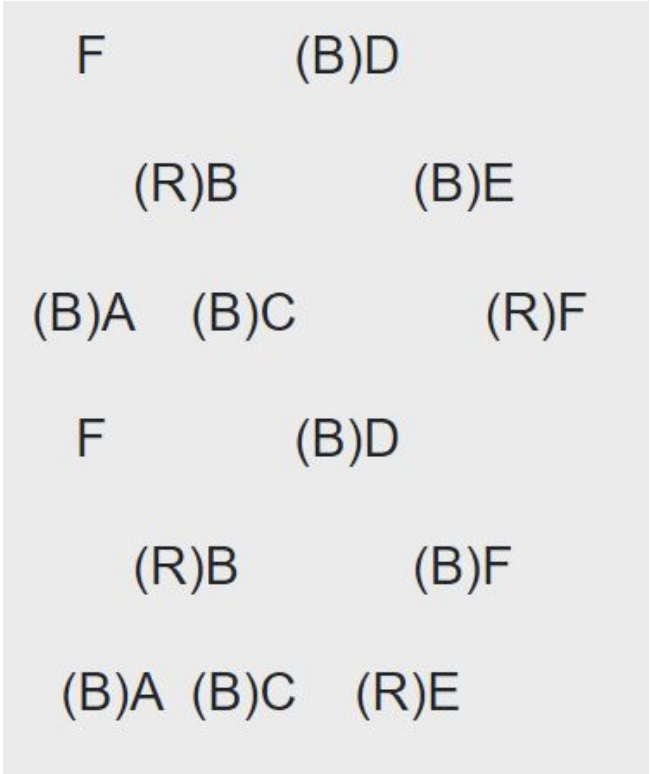




Answer: Insert F

Answer: Insert F

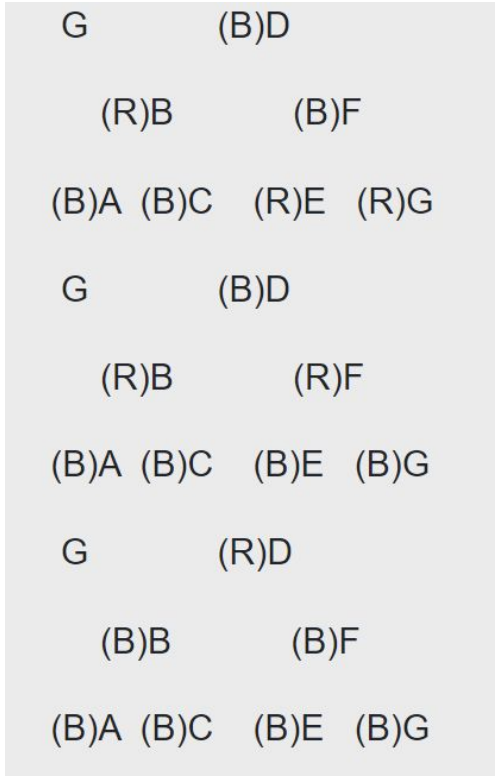
Insert →



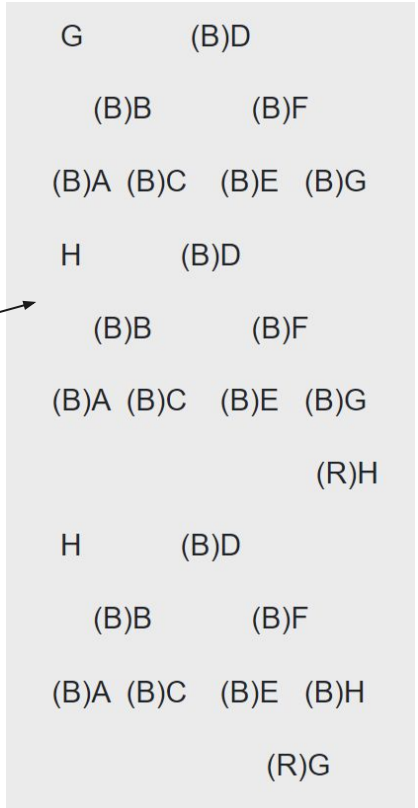
Answer: Insert G, H

Answer: Insert G, H

Insert



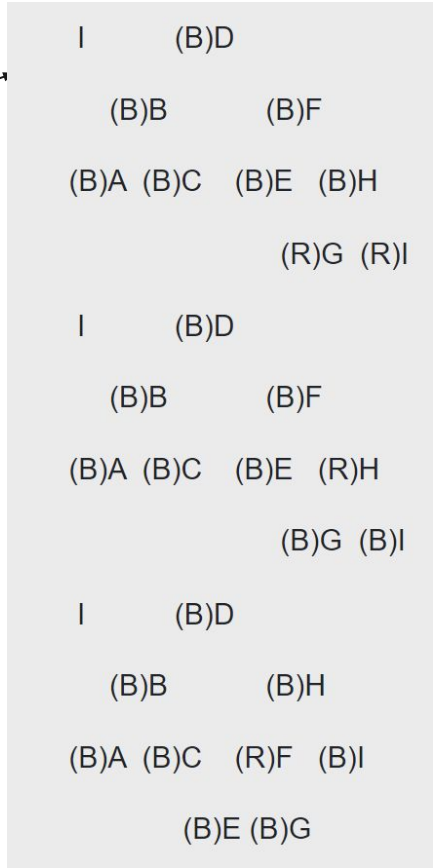
Insert



Answer: Insert I

Answer: Insert I

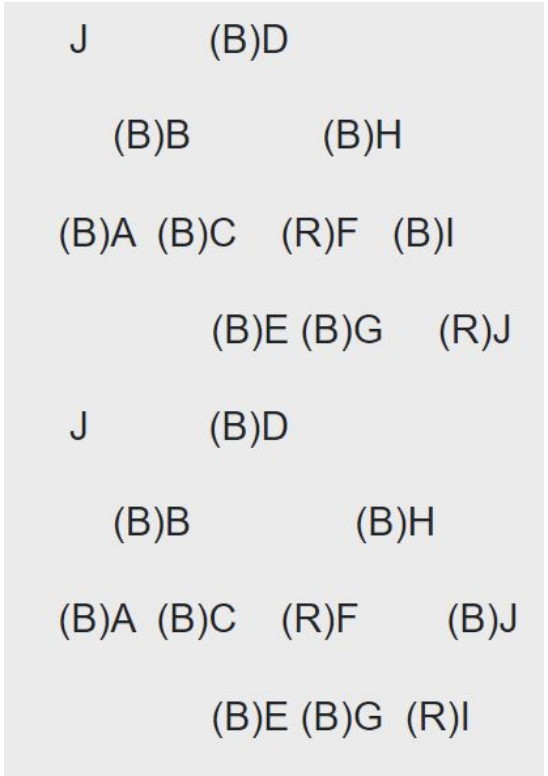
Insert



Answer: Insert J

Answer: Insert J

Insert

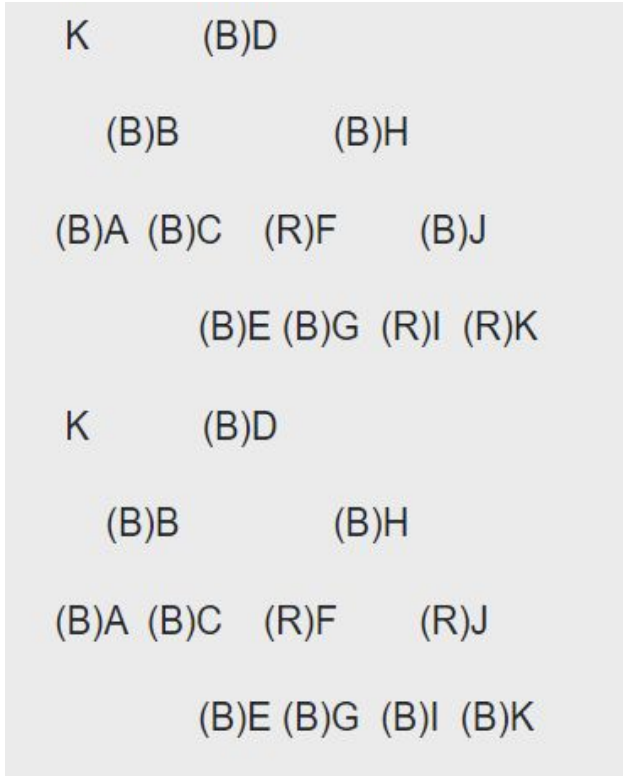


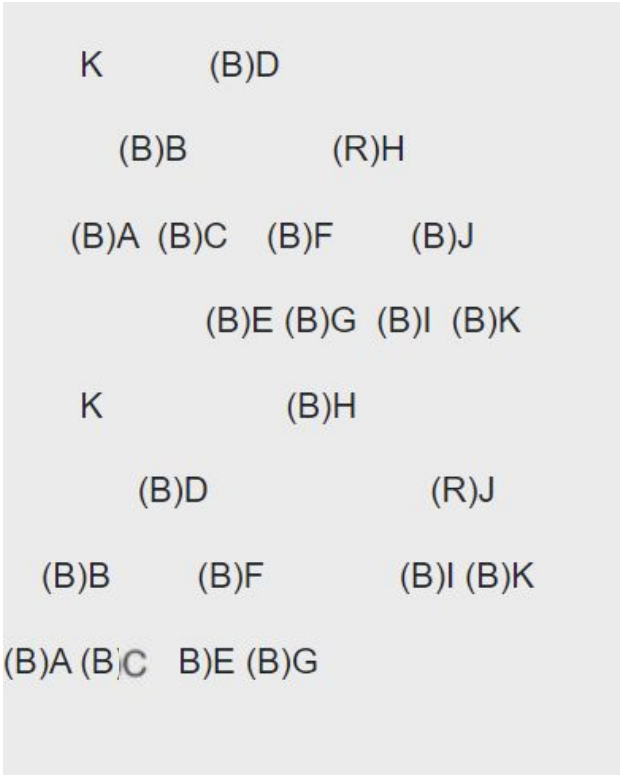
Answer: Insert K

C

Answer: Insert K

Insert





RB - Insert Ascending Order Answer

- Every insert operation requires either coloring (zero or more operations), left rotation (at most one) or a combination of coloring and left rotation
- The more nodes in the tree, the more nodes involved in these operations
- The tree height is monotonically increasing when inserting keys in increasing order.

Comparing RB and Unbalanced BSTs

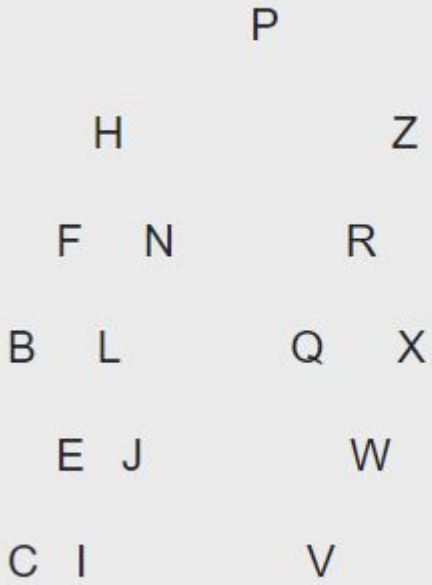
Create both the RB and unbalanced BST with the following insertion order of keys. What is the worst case runtime of searching in both?

Keys: P Z R X W H F N B L J I E Q C V

Answer

Red-black BST built with random keys 1:

BST built with random keys 1:



Answer

What is the worst case runtime of searching in both?

- _____ for the RB tree
- _____ for the BST

Good Work!

Go to https://dynrec.cs.rutgers.edu/live/

Enter the Quiz Code: