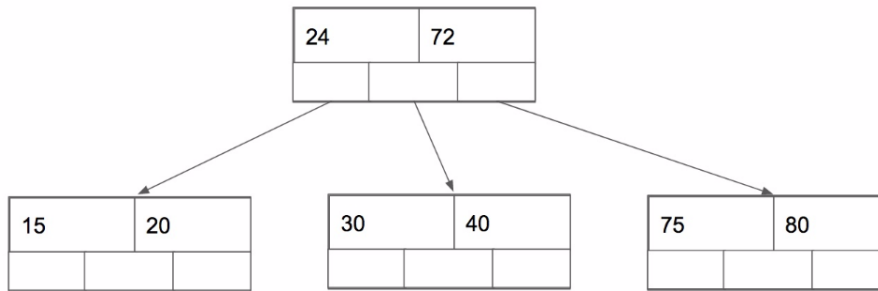


1 All about Trees

1. Why does a binary search tree have a worst case runtime of  $O(n)$  for *find*?
2. Give a sequence of operations, such that if they were inserted in the order they appear, would result in a "poor" binary search tree.
3. Examine this B-tree with order 3. Mark the paths taken when the user calls *find*(40).



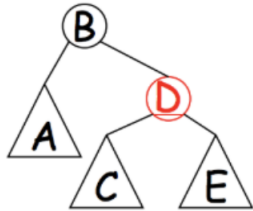
4. Now call *insert*(35), and draw the resulting tree.
5. What property of a B-tree rectifies problems of binary search trees, such as the one in 1.1? Why would you not use a B-tree?

## 2 The Holy LLRB Invariant

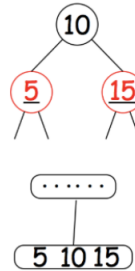
**RB Tree Invariants:** Node labels are in order from left to right. All paths through the tree contain the same number of black nodes. No red nodes have red parents. As a result, the height of a RB tree with  $n$  nodes is  $O(\log n)$ .

LLRB trees must also maintain the following invariant (in addition to the regular red-black invariant):

No right-leaning trees (black parent with right red child):



No "4-nodes" (black parent with two red children):



1. What are the "fixups" for the two cases above in order to preserve the LLRB invariant (i.e. what operations do we perform on each tree to ensure it is a proper LLRB)?

Consider the following RB tree:



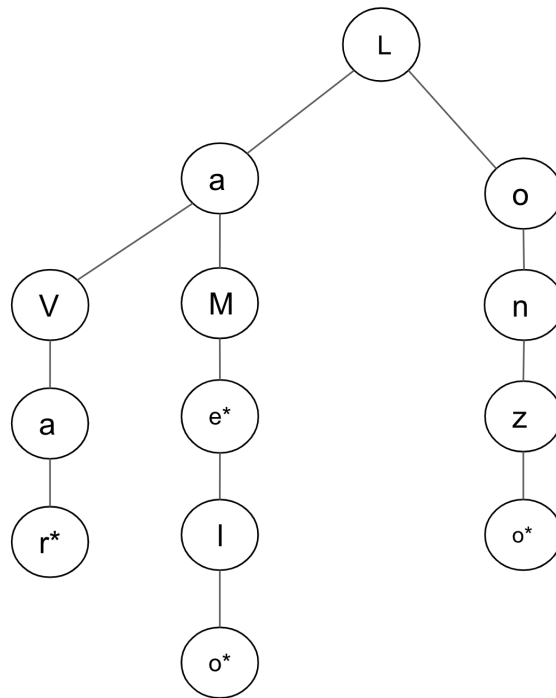
2. Draw the tree after applying all necessary fixups to make it a proper LLRB tree.

3. Next, insert 10 into the tree, and apply all fixups to preserve the LLRB invariant.

4. Finally, draw the corresponding 2-3 tree.

### 3 Trie Me

The Big Baller Brand has decided to use a trie to have fast lookup of their Big Ballers. Currently, the state of the trie is as follows:



1. The Biggest Baller of them all, CEO LaVar Ball, enjoys being reminded of who is a Big Baller. Remind him of who the Big Ballers are by finding all the words in the trie. Note: The nodes with an asterisk denote the end of a word.
2. Not again! LaVar Ball has forgotten about his son LiAngelo once again. Help LaVar by inserting "LiAngelo" and "Love" into the trie above so that no Big Baller is forgotten.
3. How long does it take to add  $n$  words, each of max length  $L$ ?
4. What's the best and worst case runtime to check whether a word of length  $L$  is in the trie?