Review Sheet 8a

CS 70: Data Structures and Program Development

Tuesday, March 9, 2020

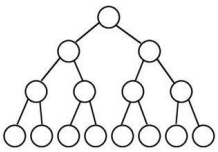
Learning Targets

1. I can explain the fundamental idea behind Red-Black trees.
2. I can explain the fundamental idea behind 2-3-4 trees.
3. I can explain the fundamental idea behind Red-Black trees.

Review

1. A *Perfect* Tree

   All levels are full; $2^{h+1}-1$ elements, where h is height.



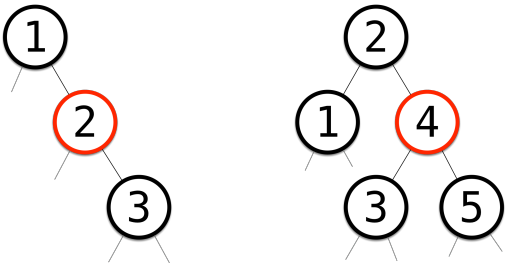
2. Red-Black Trees (definition 1)

   A red-black tree is a BST such that:

   - Every node is red or black
   - (Optional) The root is black
   - No red parent has a red child
   - Every path from the root to `nullptr` passes through the same number of black nodes.

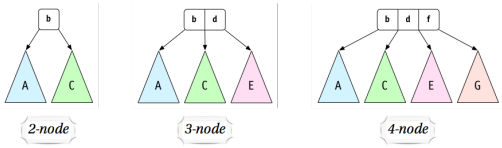
   An n-node red-black tree height $O(\log n)$

3. Valid Red-Black Trees?



4. Why not require perfection at all times?

5. Nodes in a 2-3-4 Tree



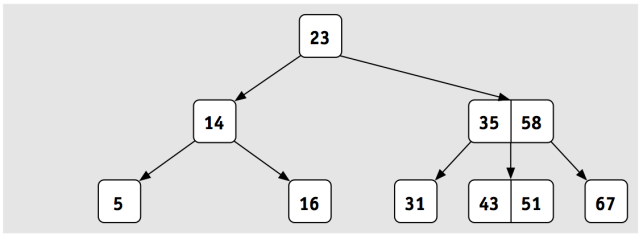
6. Insert 1, 2, 3, 4, 5, 6, ... into an empty 2-3-4 tree.

7. Advantages of 2-3-4 Trees

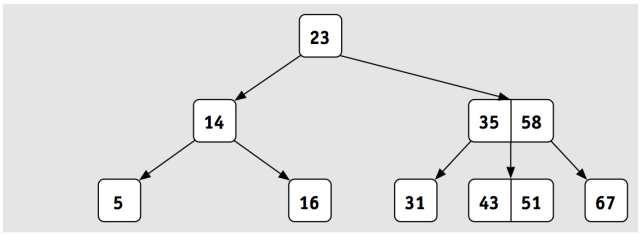
   - The tree is always "balanced"
   - Worst case for insert and lookup is $O(\log n)$ for a tree with n nodes
   - Simple algorithm; no rotations required.
   - Smaller height than a typical binary tree. (why?)

   Disadvantages?

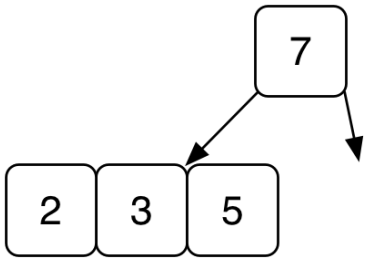
8. Exercise: Convert to Binary Tree + Superlinks



9. Exercise: Convert this tree



10. Promote the 3



11. How about now? Promote the 3