3/1 Activities

Red-Black Trees  
Red-Black Properties

1. Every node is colored either red or black
2. The root is black
3. Every null pointer descending from a leaf is considered to be a null black leaf node
4. If a node is red, then both its children are black
5. For each node, all paths from the node to descendant leaves contain the same number of black nodes (black height)

black-height of a node bh(x) - The number of black nodes on any path from, but not including, a node "x" down to a null black leaf node

1. If a node x has bh(x)=3, what is its largest and smallest possible height (distance to farthest leaf) in the BST?

2. Prove using induction and red-black tree properties. A red-black tree with n internal nodes (n key values) has height at most 2lg(n+1)

Red-Black Tree Insert - Similar to BST insert, assume we start with a valid red-black tree.

1. Locate leaf position to insert new node
2. Color new node red and create 2 new black nil leafs below newly inserted red node
3. If the parent of the new insert was \_\_\_\_\_\_\_\_(fill in the blank, black or red), then done. ELSE procedure to recolor nodes and perform rotations to maintain red-black properties.

There are three cases if R-B Property #4 broken when inserting a red node "Z" (or changing the color of a node to red) and its parent is also red.

|  | Case 1. |
| --- | --- |
|  | Case 2. |
|  | Case 3. |