1. (15%) What is the maximum black height and actual height of a RB Tree with 15 nodes? Draw a RB tree with 15 nodes that can achieve this maximum height. **(Draw and explain your answer.)**

Answer:

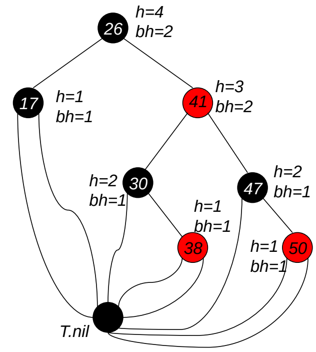
- Height of a node:
  - $h(x)$ = number of edges in a longest path to a leaf.
- Black-height of a node x, $bh(x)$:
  - $bh(x)$ = number of black nodes (including *T.nil*) on the path from x to leaf, not counting x.
- How are they related?
  - $bh(x) \le h(x) \le 2\, bh(x)$
- Lemma: The subtree rooted at any node x has $\ge 2^{bh(x)}-1$ internal nodes.
- **Lemma 13.1:** A red-black tree with n internal nodes has height at most $2 \lg(n+1)$.

An **internal node** or inner **node** is any **node of a tree** that has child **nodes** and is thus not a leaf **node**. A nonleaf node is an internal node. (Page 1176 of Textbook)

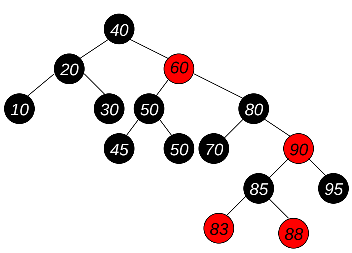
H3.1

Height of a Red-black Tree

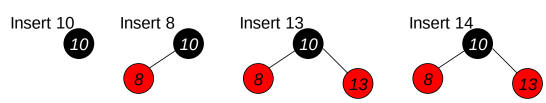
- Example:
- Height of a node:
  $h(x)$ = # of edges in a longest path to a leaf.
- Black-height of a node $bh(x)$ = # of black nodes on path from x to leaf, not counting x.
- How are they related?
  - $bh(x) \le h(x) \le 2\, bh(x)$



- Lemma: The subtree rooted at any node x has $\ge 2^{bh(x)}-1$ internal nodes.
- $14 \ge 2^{bh(x)}-1 \Rightarrow 15 \ge 2^{bh(x)} \Rightarrow \log_2(15) \ge bh(x) \Rightarrow 3.907 \ge bh(x) = 3$
- Therefore, the maximum black height is 3
- ***$bh(x) \le h(x) \le 2\, bh(x) \Rightarrow h(x) \le 2\, bh(x) \Rightarrow h(x) \le 2 \times 3 = 6$ (including leaf node)***
- Therefore, the maximum actual height is 5 (excluding leaf node)

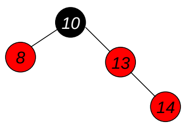


2: (30%) Follow the RB-Insert and RB-Insert-fixup pseudo codes in the textbook (Ch13.3 page 315 and 316) to insert the following values:
10, 8, 13, 14, 18, 11, 15, 17
into an initially empty RB tree.
To receive full points, you must draw each corresponding RB Tree for every corresponding case and every step in each case (i.e. case 0, 1, 2, 3) when inserting a value. In each step of RB tree graph, you MUST show and specify corresponding case numbers, recolors, and rotations.
(Hint: a total number of 15 RB trees must be drawn including an initial step only containing root node 5)

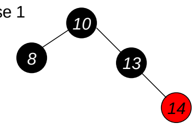


10, 8, 13, 14, 18, 11, 15, 17

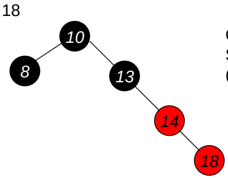
Insert 14



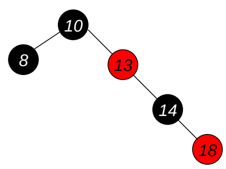
Case 1



Insert 18

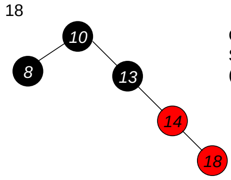


Case 3
Step 1
(recolor)

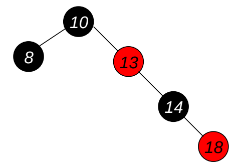


10, 8, 13, 14, 18, 11, 15, 17

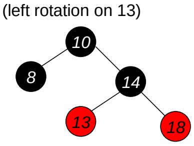
Insert 18



Case 3
Step 1
(recolor)

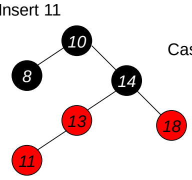


Case 3
Step 2
(left rotation on 13)

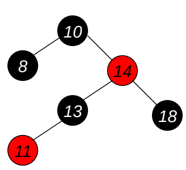


10, 8, 13, 14, 18, 11, 15, 17

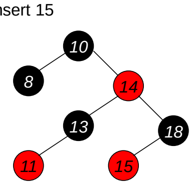
Insert 11



Case 1

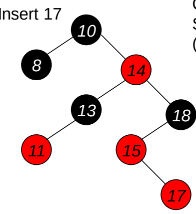


Insert 15

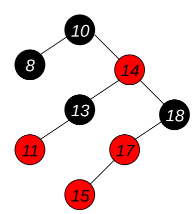


10, 8, 13, 14, 18, 11, 15, 17

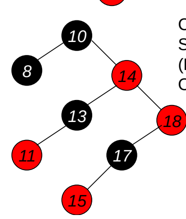
Insert 17



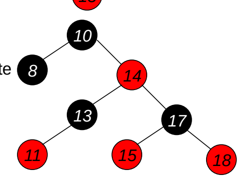
Case 2
Step 1
(left rotation on 15)



Case 2
Step 2
(Recolor)



Case 3
Step 1
(Right Rotate
On 18)



3: (15%) Prove the following statements regarding RB Tree:

a) "If a Black node has only one child that child must be a Red leaf."

Because the black height of the external nodes of the black child could NOT equal the black height of the other external node of the given black node. Consider the following diagram where Black nodes are denoted by B, and external nodes are denoted by nil:

```
        B
       / \
      B   nil
     / \
   nil  nil
```

When you take the path from the root of the tree to the top B, let's say you hit b Black nodes. Now, there are b+1 Black nodes to reach the two nil nodes on the left, but only b Black nodes to reach the nil node on the right.

H1.9

3: (15%) Prove the following statements regarding RB Tree:

b) "If a Red node has any children, it must have two children and they must *be Black."*

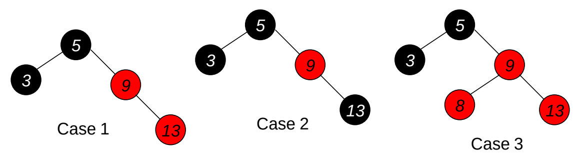
Answer:

Case 1: Node 9 is red and has only one child node 13 colored red, it violates property 5, bh on node 5

Case 2: : Node 9 is red and has only one child node 13 colored black, it violates property 5, bh on node 9

Case 3: Node 9 is red and has only two children node 8 and13, both colored red, it violates property 5, bh on node 5

Therefore, "If a Red node has any children, it must have two children and they must *be Black."*



3: (15%) Prove the following statements regarding RB Tree:

c) "Due to the property rules there are limits on how unbalanced a Red Black tree may become."

Answer:

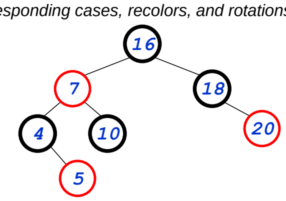
RB Tree compares to other tree structures, it has more strict rules (colored nodes) on balance of the height of subtrees, such as the relationship between Black Height and tree height.

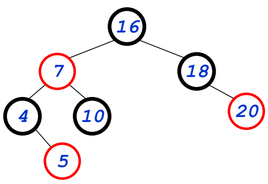
- ***Lemma:*** *The subtree rooted at any node x has $\geq 2^{bh(x)}-1$ internal nodes.*
- ***Lemma 13.1:*** *A red-black tree with n internal nodes has height at most 2 lg(n+1).*
- ***bh(x) ≤ h(x) ≤ 2 bh(x)***

H1.11

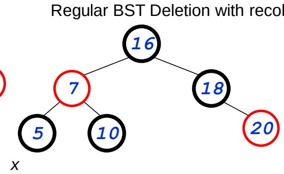
4. (25%) Follow the RB-Delete and RB-Delete-fixup pseudo codes in the textbook (Ch13.3 page 324 and 326) to delete value 4 and then delete 7 from the following RB tree.

To receive full points, you must draw each corresponding RB Tree for every corresponding case and every step in each case (i.e. case 1, 2, 3, 4) when deleting a node. In each step of RB tree graph, you MUST show and specify corresponding cases, recolors, and rotations.

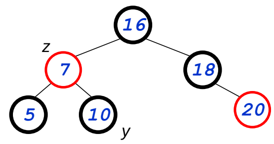




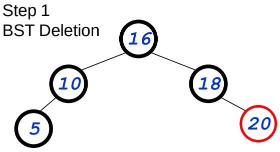
Delete 4
Case 0
Regular BST Deletion with recolor



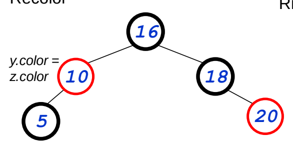
Delete 7



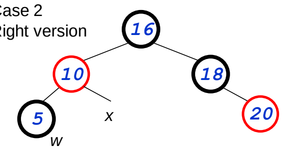
Delete 7
Step 1
BST Deletion



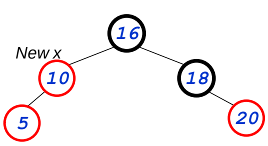
Delete 7
Step 2
Recolor



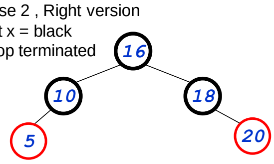
Delete 7
Step 3
Case 2
Right version



Delete 7
Step 4
Case 2 , Right version

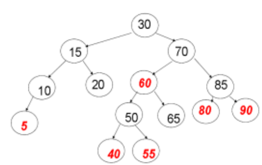


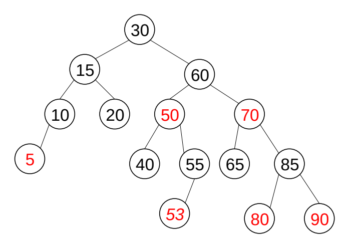
Delete 7
Step 5
Case 2 , Right version
Set x = black
Loop terminated



5. (15%) Draw the result RB trees for the following operations;

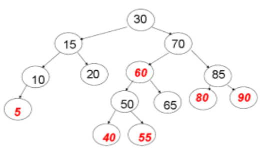
a) Using the RB Tree below, insert value 53 to the RB tree.

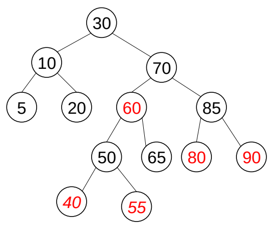




5. (15%) Draw the result RB trees for the following operations;

b) Using the RB Tree below, delete value 15 from the RB tree.





5. (15%) Draw the result RB trees for the following operations;

c) Using the RB Tree below, delete value 60 from the RB tree.