CS 385, Homework 5: Balanced Trees and Transform-and-Conquer

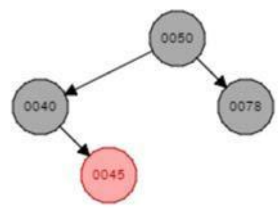
Name: Max Shi

Date: 11/21/2019
I pledge my honor that I have abided by the Stevens Honor System.

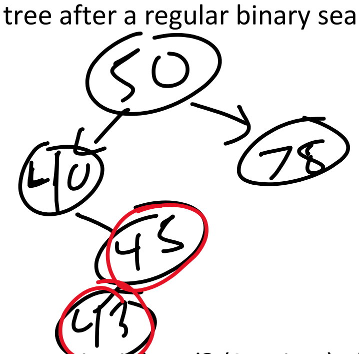
Point values are assigned for each question.

Points earned: ____ / 74, ____ %

1. Show how the red-black tree would look after inserting a node with the key 0043. Use the document on Moodle that explains the insertion process succinctly. List the case you applied (i.e. 1, 2a, 3b), and write the steps you took to fix the tree (also listed in the document).



a) Draw the tree after a regular binary search tree insertion. (3 points)

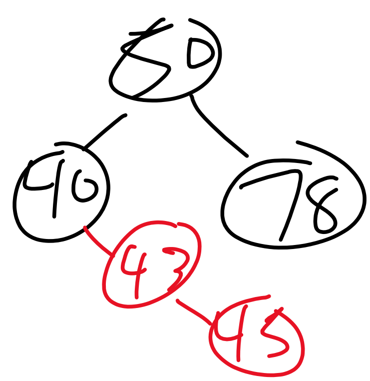


b) Which property is violated? (3 points) The height of the tree is greater than lg (n) + 1. The height of the tree is 4, and lg (5) + 1 = 2.--- + 1 = 3.---, and 4 is greater than 3.---. Also, in a regular insertion, 45 would be red, and 43 would also be inserted red, violating the principle that a red node needs both its children to be black.

Case seen after regular binary search tree insertion: (3 points)
2b

Steps taken to fix the tree: (3 points)
Right rotate the parent of the inserted node, and fixup the parent.

Draw the tree after taking the steps you just described. (3 points)

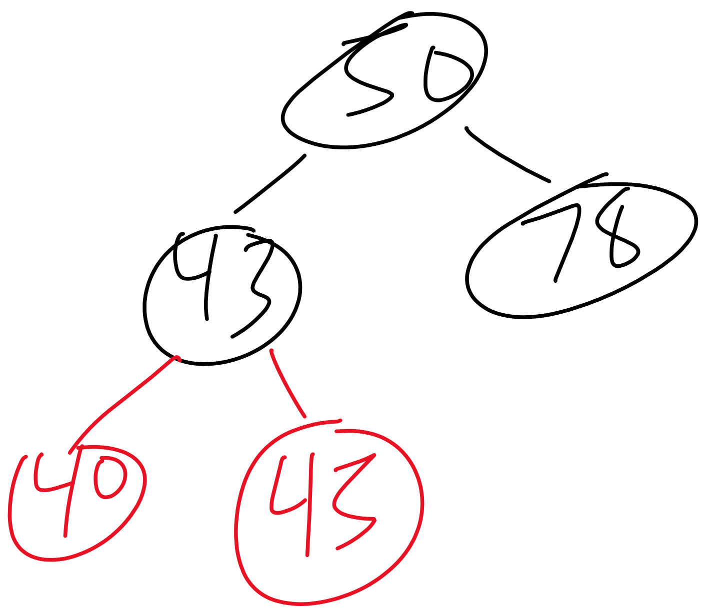


c) Which property is violated now? (3 points) The same property as before, with the height being too much as well as a red node having a red child.

d) Case seen after first fixup: (3 points) 3b

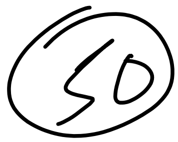
Steps taken to fix the tree: (3 points) Z is now 45. Set the parent of Z to be black, set the parent of the parent of Z to be red, and left rotate that node.

Draw the tree after taking the steps you just described. (3 points)

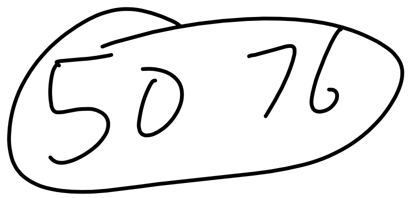


2. Draw the 2-3 tree after inserting each of the following keys. Redraw the tree for each part.

a) 50 (1 point)



b) 76 (1 point)

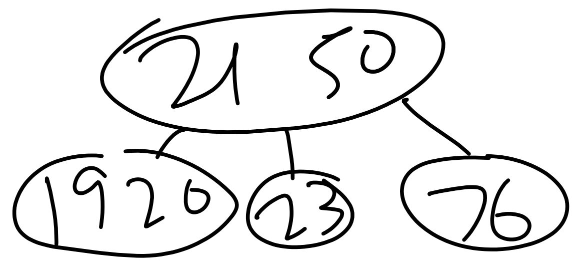


c) 23 (3 points)

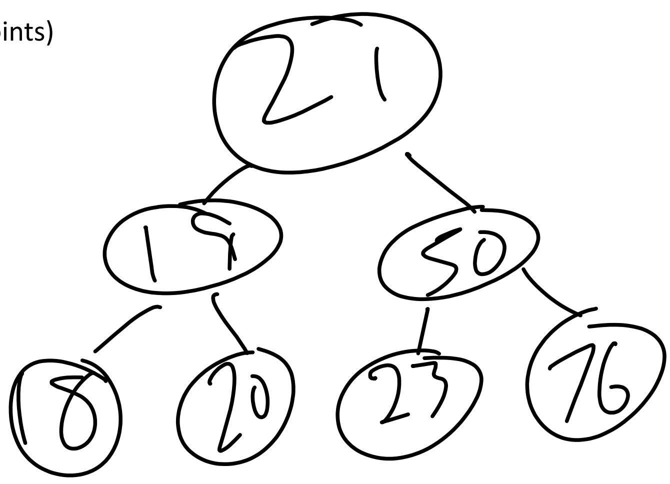
d) 21 (3 points)

e) 20 (3 points)

f) 19 (3 points)



g) 18 (3 points)



3. Read pages 241-242 in the textbook. Using that information, write pseudocode for computing the LCM of an array A[1..n] of integers. You may assume there is a working gcd() function. (6 points)

ALGORITHM *LCM*(A[1..n]):

```
// Computes the least common multiple of all the integer in array A
lcm = A[1]
for (int i = 2; i<=n; i++){
      GCD = gcd(lcm, A[i])
      lcm = lcm * A[i] / GCD
}
return lcm
```

4. Horner's method:

$4x^4 + 5x^3 - 2x^2 - 4x + 7$

a. Repeatedly factor out x in the following polynomial so that you can apply Horner's method. Write your expression for p(x). (5 points)

P(x) = 7 + x(-4 + x(-2 + x(5 + 4x)))

b. Show values of the array P[0..n] as needed to apply Horner's method. (3 points)

[7, -4, -2, 5, 4]

c. Evaluate polynomial at x = 2

x	p	n	i
2	4	4	
2	8+5 = 13		3
2	26-2 = 24		2
2	48-4 = 44		1
2	88+7 = 95		0

P(2) = 95

d. Use synthetic division to divide p(x) by x-2.

2 | 4 5 -2 -4 7
 8 26 48 88
4 13 24 44 95

5. Rewrite the *LeftRightBinaryExponentiation* algorithm on page 237 in the textbook to work for as well as any positive integer. *No credit will be given for answers that simply start with an if* 0 *statement for n = 0.* (6 points)

ALGORITHM *LeftRightBinaryExponentiation*(a, b(n)):

```
// Computes a^n
product = 1
```

```
for i = I down to 0 do
        product *= product
        if b[i] == 1 product *= a
return product
```