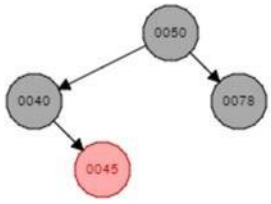
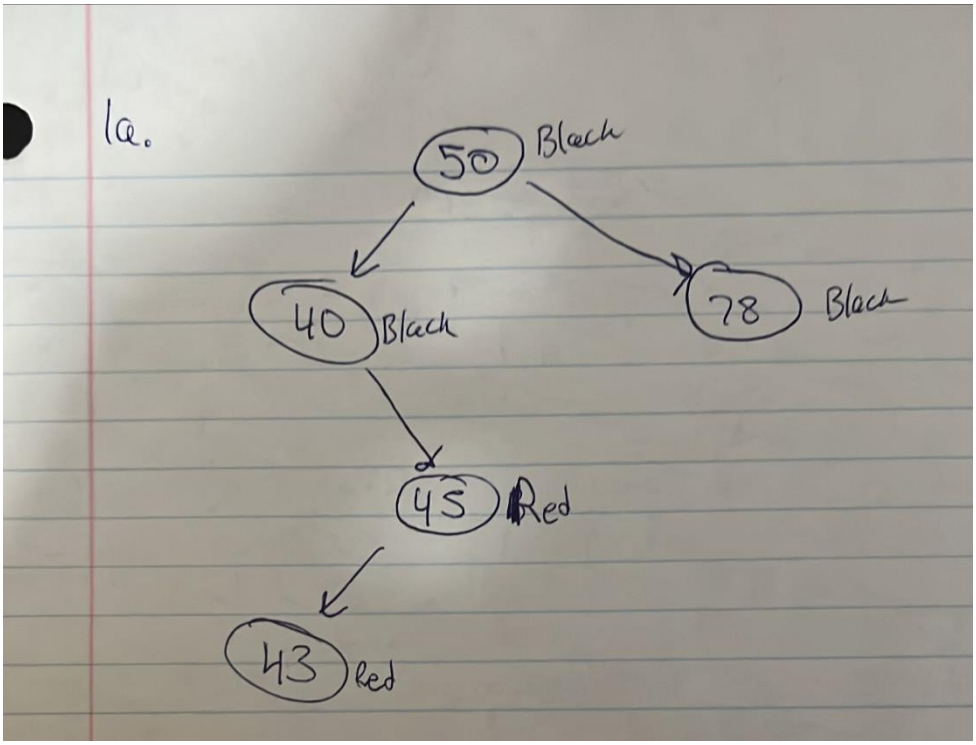
Name: Brian Sampson Date: 11/26/22

Points earned: ____ / 74 = ____ %

1. Show the red-black tree after inserting a node with the key 0043. Use the document on Canvas 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).



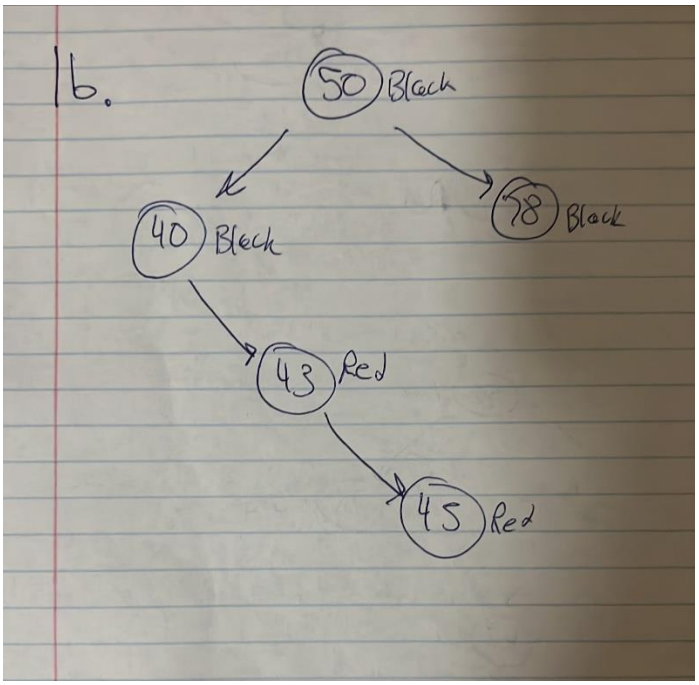

a) Draw the tree after doing a regular binary search tree insertion of 0043. (3 points)




b) Which RBT property is violated? (3 points) Since the inserted node is automatically red, property 4 is violated because of how node “0043” is red and node “0045” is also red. Property 4 states that if a node is red, then both its children are black. This is not the case in this tree.

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

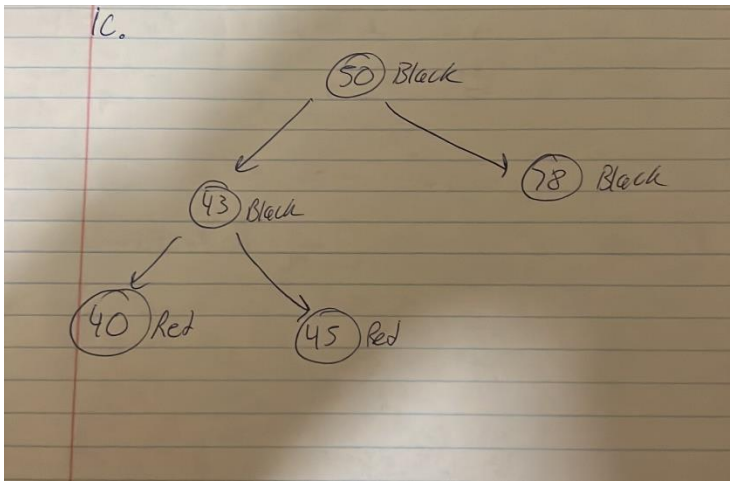
Steps taken to fix the tree: (3 points)
Let z be the inserted node.
Z = p[z] (z becomes its parent)
Right-rotate(z)
Draw the tree after taking the steps you just described. (3 points)



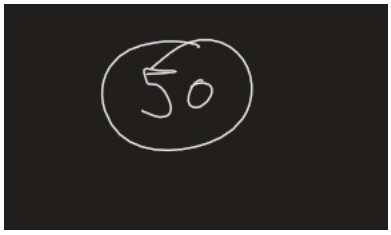

c) Which property is violated now? (3 points) Property 4 is still violated at this point because of how node “0043” and node “0045” are both red. Red nodes must have 2 black children.

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

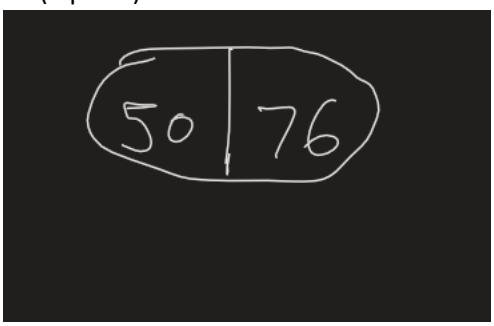
Steps taken to fix the tree: (3 points)
Z is currently the “0045” node.
P[z].color = black (make the parent of z black)
P[p[z]].color = red (make the parent of the parent of z red)
Left-rotate(p[p[z]])
Draw the tree after taking the steps you just described. (3 points)



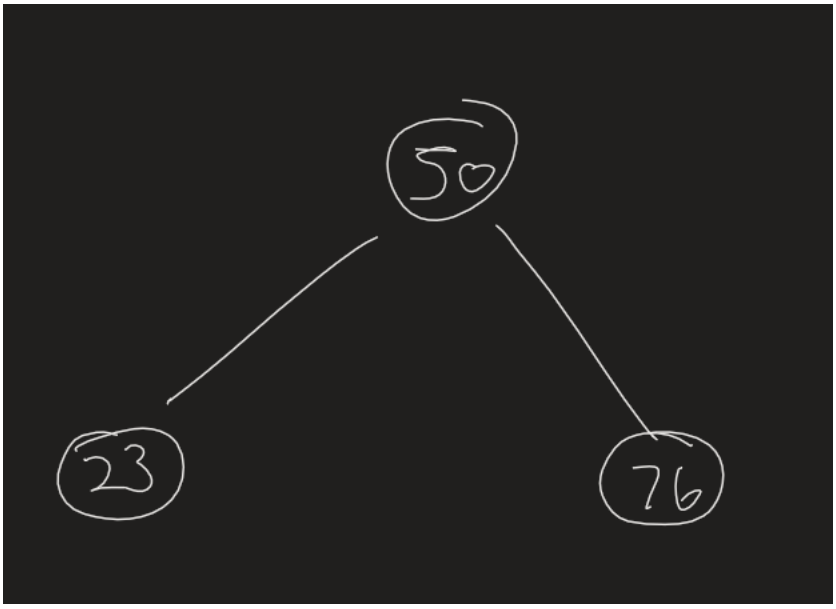

2. Draw the 2-3 tree after inserting each of the following keys. Redraw the whole tree for each part.
   a) 50 (1 point)



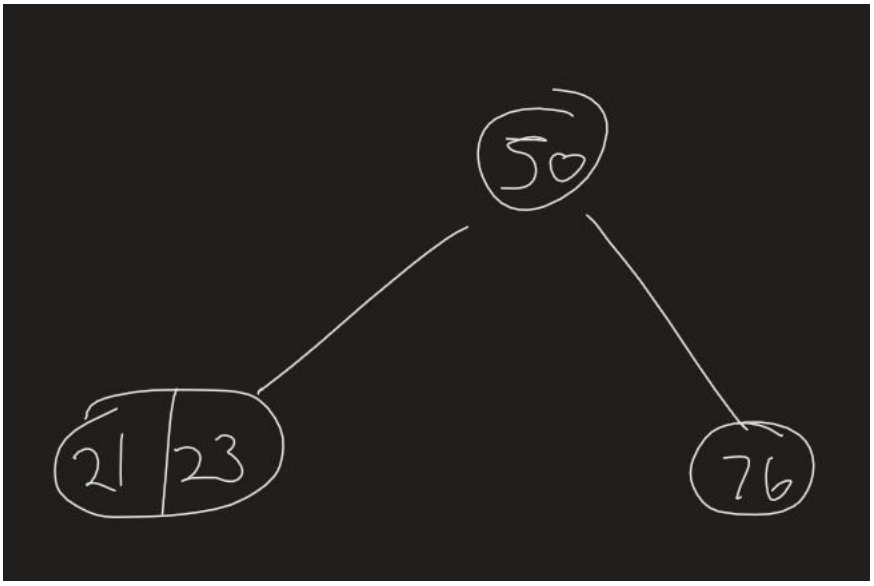

   b) 76 (1 point)



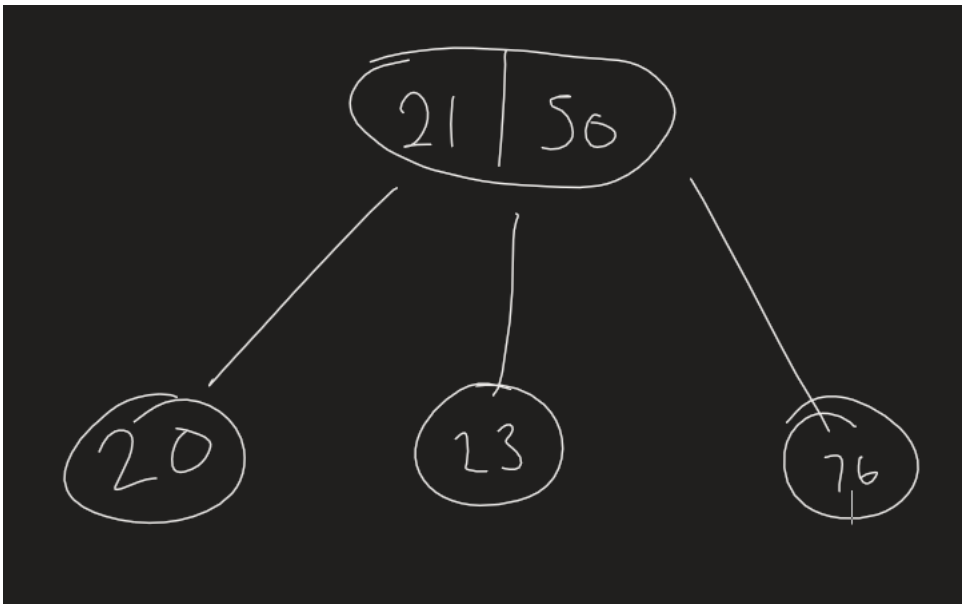

   c) 23 (3 points)



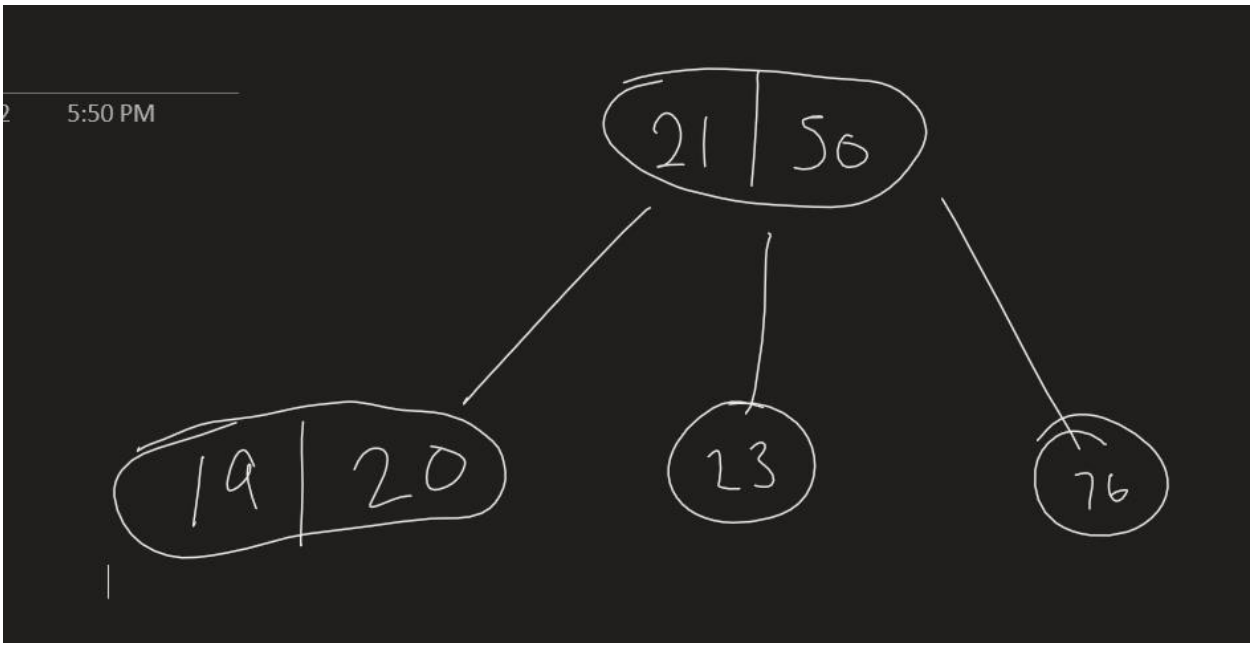

d) 21 (3 points)



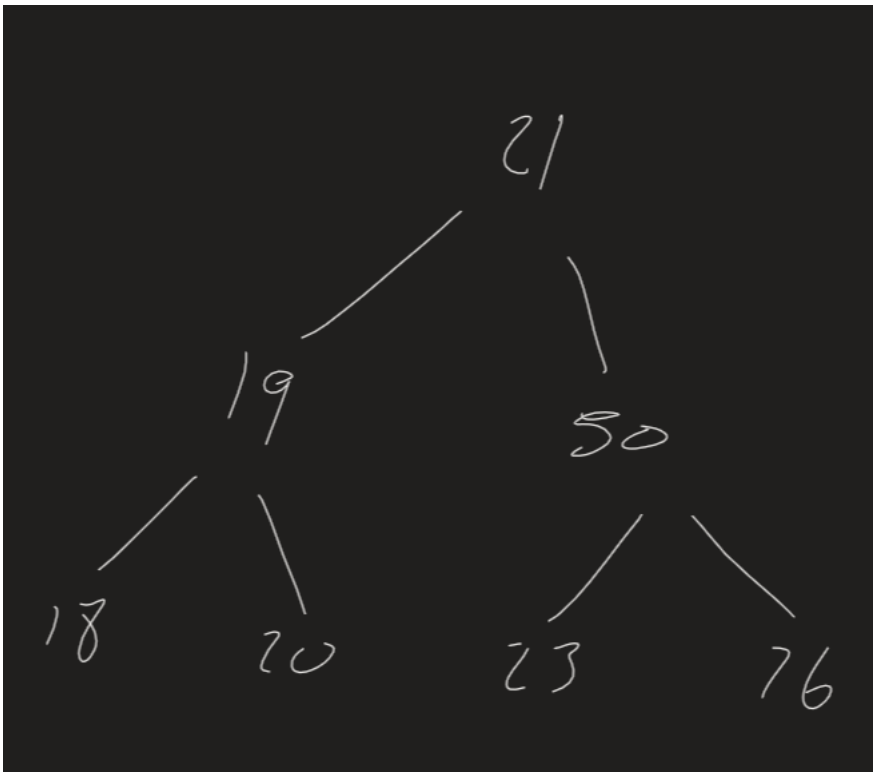

e) 20 (3 points)




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 gcd() function available. (6 points)

```
ALGORITHM LCM(A[1..n]):
    // Computes the least common multiple of all the integer in array A
    Result = A[0]
    For I in range(1, n):
            Result = result * i / gcd(result, i)
    Return result
```

4. Horner's method: $p(x) = 4x^4 + 5x^3 - 2x^2 - 4x + 7$
   a) Repeatedly factor out x in the polynomial above so that you can apply Horner's method. Write your final expression for $p(\mathrm{x})$. (5 points)
   P(x) = 7 + x(-4 + x(-2 + x(5 + x(4))))

   b) Show values of the array P[0..n] as needed to apply Horner's method. (3 points)
   [7,-4,-2,5,4]

c) Apply Horner's method to evaluate the polynomial at $x$ 2. Make a table as we did in class showing the values $x$, $p$, $n$, and $i$, and then state your final answer for $p(2)$. (5 points)

| x | p | n | i |
|---|---|---|---|
| 2 | 4 | 4 | |
| | 13 | | 3 |
| | 24 | | 2 |
| | 44 | | 1 |
| | 95 | | 0 |

$p(2) = 95$

d) Use **synthetic** (not long) **division** to divide $p(x)$ by $x$ - 2 to check your work. Be sure to show your work. (5 points)

| X0 | X4 | X3 | X2 | X1 | X0 |
|---|---|---|---|---|---|
| 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 $n = 0$ as well as any positive integer. *No credit will be given for answers that simply start with an if statement for n = 0.* (6 points)

```
ALGORITHM LeftRightBinaryExponentiation(a, b(n)):
    // Computes a^n
Product = 1
For i = I downto 0:
        Product = pow(product, 2)
        If bi = 1:
                Product = product * a
Return product
```