3013 - Algorithms
Spring 2024
Exam Two
April 1, 2024

Name: _____

Question	Points	Score
1	2	
2	2	
3	2	
4	2	
5	2	
6	2	
7	2	
8	2	
9	2	
10	2	
11	2	
12	2	
13	2	
14	2	
15	2	
16	2	
17	2	
18	2	
19	2	
20	2	
21	2	
22	2	
23	2	
24	2	
25	2	
26	2	
27	2	
28	20	
29	10	
30	10	
31	10	
32	15	
33	30	
34	7	
Total:	156	

1. Multiple Choice

1.1. Binary Trees

1. (2 points) What is a defining property of a Binary Search Tree (BST)?
   A. Each node has exactly two children.
   B. The left subtree of a node contains only nodes with keys less than the node's key.
   C. Every node has a unique color, either red or black.
   D. The nodes are always arranged in a complete binary tree.

2. (2 points) It is best to use an array to store a binary tree because:
   A. The random access arrays provide is fast.
   B. Indexing the nodes provides a database style lookup (direct access).
   C. Calculating children using an index is faster than following pointers,
   D. Its not better.

3. (2 points) What is the time complexity of searching for a value in a Binary Search Tree (BST)
   A. O(1)
   B. O(log n)
   C. O(n)
   D. O(n log n)

4. (2 points) Which of the following statements is true regarding the insertion of a new node in a BST?
   A. The new node is always added as a new root.
   B. The new node is always added as a leaf.
   C. The new node can replace an existing node if they have the same value.
   D. The new node is always added to the left subtree.

5. (2 points) When deleting a node with two children from a BST, which of the following strategies is often used to find a replacement node?
   A. Selecting the smallest node from the right subtree.
   B. Selecting the largest node from the left subtree.
   C. Selecting any leaf node.
   D. Both a) and b) are correct.

6. (2 points) Which of the following operations can potentially violate the BST property if not done correctly?
   A. Searching for a node
   B. Rotating the tree to balance it
   C. Traversing the tree in-order
   D. Modifying the value of a node

7. (2 points) When deleting a node from a BST, which case is the easiest?
   A. Deleting the root
   B. Deleting siblings
   C. Deleting a node with no descendants
   D. Deleting a node with one child

8. (2 points) When deleting a node with only one child in a BST, what happens to the child?
   A. It is also deleted.
   B. It replaces its parent in the tree structure.
   C. It becomes a leaf node.

D. Level-order traversal

17. (2 points) For which of the following applications would an in-order traversal of a binary search tree be most appropriate?
    A. To delete the tree
    B. To clone the tree
    C. To print the elements of the tree in sorted order
    D. To search for an element in the tree

18. (2 points) Post-order traversal is particularly useful for:
    A. Printing a binary tree in ascending order.
    B. Deleting or freeing nodes and resources of the tree from the bottom up.
    C. Copying the tree.
    D. Implementing infix expressions.

1.2. Recursion

19. (2 points) What is the base case in a recursive algorithm?
    A. The condition that reduces the problem space.
    B. The condition that terminates the recursive calls.
    C. The first call to the recursive function.
    D. The final result of all recursive calls.

    Consider the following Python function:

```python
def print_numbers(n):
    if n > 0:
        print(n)
        print_numbers(n-1)

```

20. (2 points) What will print_numbers(3) output?
    A. 1 2 3
    B. 3 2 1
    C. 3 2 1 0
    D. An error message

21. (2 points) A recursive function must always have which of the following?
    A. At least two recursive calls
    B. A while loop
    C. A base case
    D. A global variable

22. (2 points) A recursive function does what to ensure it will reach the base case and finish.
    A. Call itself
    B. Use Memoization to speed up calculations.
    C. Reduce the problem space
    D. Use the static keyword to ensure every instance of the function has access to0 the same variables

23. (2 points) A recursive function is always more efficient than an iterative function:
    A. True
    B. False

D. It is moved to the opposite subtree to maintain balance.

9. (2 points) When deleting a node with two children, the successor node is often used to replace the deleted node. How is the successor node defined?
   A. It is the smallest node in the left subtree.
   B. It is the largest node in the right subtree.
   C. It is the smallest node in the right subtree.
   D. The node in right subtree that is left heavy.

10. (2 points) Which of the following best describes the impact of deleting a node on the BST property?
   A. There is no impact, just like recursively searching for a value has no impact on the BST.
   B. Compared to other operationos, this one has the biggest change of violating the BST property if not done correctly.
   C. The BST property is irrelevant to node deletion.
   D. The BST property is maintained as long as the tree is rebalanced afterward.

11. (2 points) What is the time complexity of deleting a node in a balanced BST?
   A. O(1)
   B. O(log n)
   C. O(n)
   D. O(n log n)

12. (2 points) What is the main purpose of performing rotations in a binary search tree?
   A. To make the tree complete and balanced.
   B. To ensure that each node has at most one child.
   C. To maintain the binary search tree property after insertions or deletions.
   D. To reduce the height of the tree, thereby improving search, insertion, and deletion operations' efficiency.

13. (2 points) In an AVL tree, what type of rotation is used to fix an imbalance caused by an insertion into the left child of the left subtree of a node?
   A. Right rotation
   B. Left rotation
   C. Left-right rotation
   D. Right-left rotation

14. (2 points) Which of the following statements is TRUE about tree rotations?
   A. Rotations can violate the binary search tree property.
   B. A single rotation is always sufficient to balance a tree.
   C. Rotations are only necessary in AVL trees, not any other tree data structure.
   D. Rotations preserve the in-order traversal of the tree.

15. (2 points) When performing a right rotation on a binary search tree, what happens to the left child of the pivot node?
   A. It becomes the right child of the pivot node.
   B. It becomes the parent of the pivot node.
   C. It is unchanged.
   D. It becomes the left child of the right child of the pivot node.

16. (2 points) Which tree traversal method visits the root node before its children?
   A. In-order traversal
   B. Pre-order traversal
   C. Post-order traversal

1.3. Complexity

24. (2 points) Choose the fastest big oh complexity below:

    A. $O(1024)$

    B. $O(n)$

    C. $O(n\ log n)$

    D. $O(n^2)$

25. (2 points) Complexity of any algorithm is always based on:

    A. The number of instructions necessary to implement the algorithm.

    B. The data container it uses, since array are faster, algorithms that use arrays will be faster.

    C. The type of loop it chooses (e.g. For loops are faster since they have a definitive stopping point).

    D. The size of the data that it processes.

26. (2 points) Considering a recursive algorithm that divides the problem space in half at each step, what is its time complexity?

    A. $O(1)$

    B. $O(\log n)$

    C. $O(n)$

    D. $O(n^2)$

27. (2 points) What is the worst-case time complexity of searching for an element in an unbalanced binary search tree (BST)?

    A. $O(1)$

    B. $O(\ log n)$

    C. $O(n)$

    D. $O(n\ log\ n)$

2. Matching

28. (20 points) Use the table below to answer this amazing question:

	Term		Definition
1	Ancestor	**A**	A node directly connected to another node when moving away from the root.
2	Balanced	**B**	The maximum number of nodes at any level in the tree.
3	Child	**C**	The node with the smallest key greater than the key of the given node.
4	Depth (of a node)	**D**	The number of edges on the longest downward path between that node and a leaf.
5	Descendant	**E**	A node reachable by repeated proceeding from child to parent.
6	Height (of a node)	**F**	The number of edges from the root to the node.
7	Inner Node	**G**	The depth of a node plus one.
8	Inorder Predecessor	**H**	The height difference between the left and right subtrees is no more than one.
9	Inorder Successor	**I**	Nodes that share the same parent.
10	Leaf	**J**	A sequence of nodes and edges connecting a node with a descendant.
11	Level	**K**	The converse notion of a child.
12	Node	**L**	An individual element of a tree that contains data.
13	Parent	**M**	A node reachable by repeated proceeding from parent to child.
14	Path	**N**	A node that has at least one child (i.e., it is not a leaf).
15	Root	**O**	The node with the largest key smaller than the key of the given node.
16	Sibling	**P**	The topmost node of a tree, which has no parent.
17	Width	**Q**	A node with no children.

1. **E** (*A node reachable by repeated proceeding from child to parent.*)
2. **H** (*The height difference between the left and right subtrees is no more than one.*)
3. **A** (*A node directly connected another node when moving away from the root.*)
4. **F** (*The number of edges from the root to the node.*)
5. **M** (*A node reachable by repeated proceeding from parent to child.*)
6. **D** (*The number of edges on the longest downward path between that node and a leaf.*)
7. **N** (*A node that has at least one child (i.e., it is not a leaf).*)
8. **O** (*The node with the largest key smaller than the key of the given node*)
9. **C** (*The node with the smallest key greater than the key of the given node.*)
10. **Q** (*A node with no children.*)
11. **G** (*The depth of a node plus one.*)
12. **L** (*n individual element of a tree that contains data.*)
13. **K** (*The converse notion of a child.*)
14. **J** (*A sequence of nodes and edges connecting a node with a descendant.*)
15. **P** (*The topmost node of a tree, which has no parent.*)
16. **I** (*Nodes that share the same parent.*)
17. **B** (*The maximum number of nodes at any level in the tree.*)

3. Short Answer

29. (10 points) Can you write a recursive function that implements a countdown timer? I want your function to be called countdown, and I want the output to look like:

```
./countdown 5
Output:
5
4
3
2
1
Blastoff!!
```

Answer:

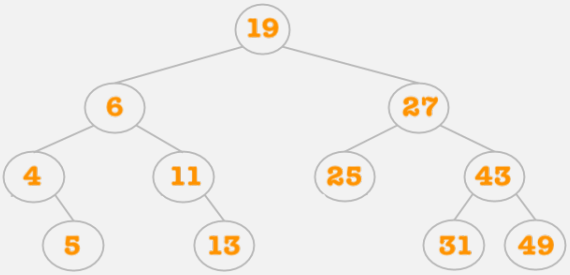
```
void countdown(int n) {
  if (n <= 0) {
    cout << "Blast off!";
  } else {
    cout << n << endl;
    countdown(n - 1);
  }
}

```

30. (10 points) Given the following values read them from left to right and insert them into a binary search tree.

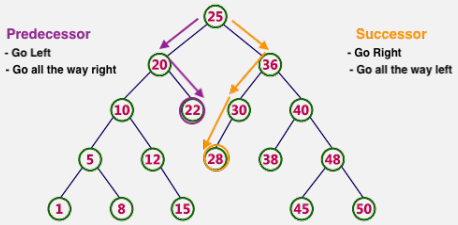
$$19,\ 6,\ 11,\ 4,\ 13,\ 5,\ 27,\ 43,\ 49,\ 31,\ 25$$

Answer:

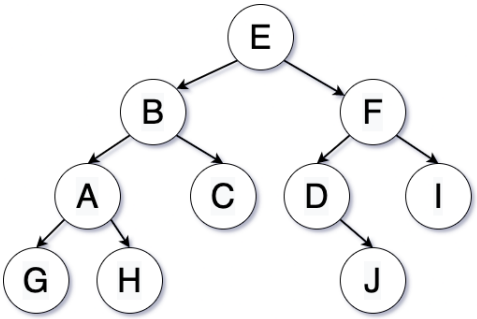


31. (10 points) Given the following tree, what are the inorder successor and predecessor of the root node?

Answer:



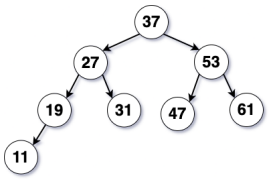
32. (15 points) Give a Pre-order / Post-order / In-order traversal of the following tree.



Answer:

- `PreOrder: E, B, A, G, H, C, F, D, J, I`
- `InOrder: G, A, H, B, C, E, D, J, F, I`
- `PostOrder: G, H, A, C, B, J, D, I F, E`

33. (15 points) Given the following AVL tree:

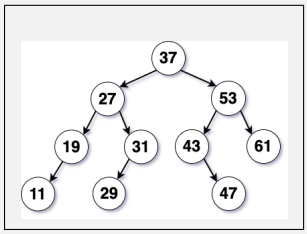


Insert the values as stated below and draw the resulting tree. Explain in short sentences what you had to do to ensure it kept its BST properties.

(a) (5 points) Insert the value **29** to this AVL tree,

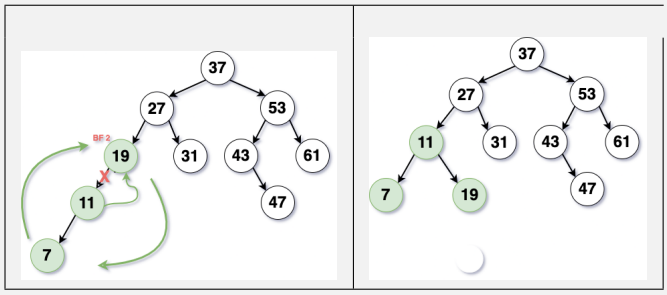
Answer:

Note: The tree representing the answer has one extra value in the right subtree (**47**), just ignore it. I removed the **43** and replaced it with the **47**. The result images took a little while to make, and I didn't have time to remake them.



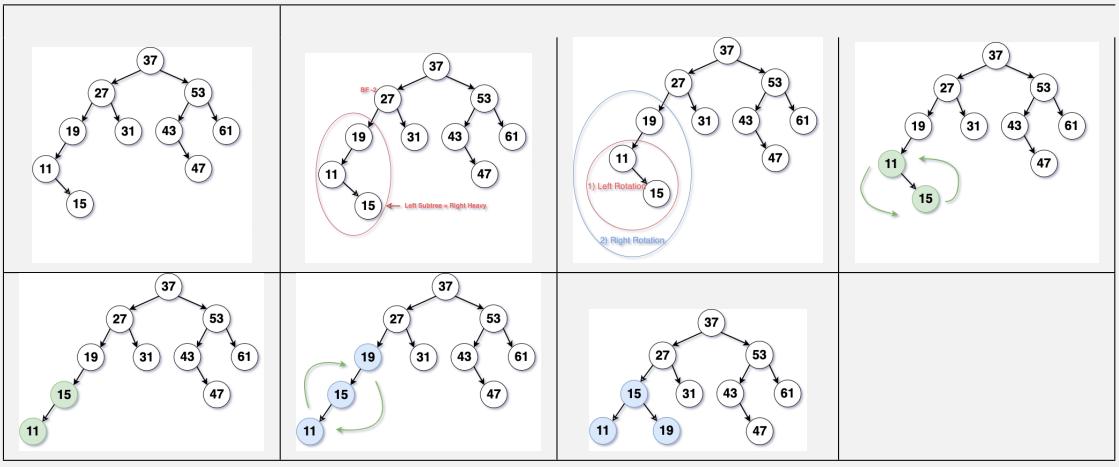
(b) (5 points) Insert the value **7** to this AVL tree.

Answer:



(c) (5 points) Insert the value **15** to this AVL tree.

Answer:



34. (7 points) **Your Choice.** What did you study that I didn't put on the exam?