

Self Test - Binary Trees

1. A binary tree is a search tree if
 - (a) every non-leaf node has children whose key values are less than (or equal to) the parent.
 - (b) every left child has a key less than the parent and every right child has a key greater than (or equal to) the parent.**
 - (c) in the path from the root to every leaf node, the key of each node is greater than (or equal to) the key of its parent.
 - (d) a node can have a maximum of two children.
2. A subtree of a binary tree always has
 - (a) a root that is a child of the main tree's root.
 - (b) a root unconnected to the main tree's root.
 - (c) fewer nodes than the main tree.**
 - (d) a sibling with the same number of nodes.
3. Finding a node in a binary search tree involves going from node to node, asking
 - (a) how big the node's key is in relation to the search key.**
 - (b) how big the node's key is compared to its right or left children.
 - (c) what leaf node we want to reach.
 - (d) what level we are on.
4. An unbalanced tree is

- (a) in which most of the keys have values greater than the average.
 - (b) whose behavior is unpredictable.
 - (c) **in which the root or some other node has many more left children than right children, or vice versa.**
 - (d) that is shaped like an umbrella.
5. In a binary tree used to represent a mathematical expression, which of the following is not true?
- (a) **Both children of an operator node must be operands.**
 - (b) Following a postorder traversal, no parentheses need to be added.
 - (c) Following an inorder traversal, parentheses must be added.
 - (d) In pre-order traversal a node is visited before either of its children.
6. Fill in the blanks
- (a) Inserting a node starts with the same steps as **finding** a node.
 - (b) Suppose a node A has a successor node S. Then S must have a key that is larger than **A** but smaller than or equal to **its parents node.**
 - (c) If a tree is represented by an array, the right child of a node at index n has an index of **$2*n+1$.**
 - (d) The number of edges from the root to the node is called **depth** of the tree.
7. State whether the following statements are TRUE or FALSE. Justify your answer.
- (a) Deleting a node with one child from a binary search tree involves finding that node's successor. **(False)**
 - (b) Not all trees are binary trees. **(True)**
 - (c) In Binary Search Tree the left and right sub-trees should also be binary search trees. **(True)**
8. What does the following piece of code do?
- ```
public void func(Tree root)
```

```
{ func(root.left());
 func(root.right());
 System.out.println(root.data());
}
```

- (a) Preorder traversal
  - (b) Inorder traversal
  - (c) Postorder traversal**
  - (d) Level order traversal
9. The preorder traversal sequence of a binary search tree is 30, 20, 10, 15, 25, 23, 39, 35, 42. Which one of the following is the postorder traversal sequence of the same tree?
- (a) 10, 20, 15, 23, 25, 35, 42, 39, 30
  - (b) 15, 10, 25, 23, 20, 42, 35, 39, 30
  - (c) 15, 20, 10, 23, 25, 42, 35, 39, 30
  - (d) 15, 10, 23, 25, 20, 35, 42, 39, 30**
10. Postfix expression for  $(A+B) * (C+D)$  is
- (a)  $A B C * + D +$
  - (b)  $A B + C D + *$**
  - (c)  $ABCD++*$
  - (d) None

\*\*\*\*\*