Q1. The pre-order traversal sequence of a binary search tree is:

27, 21 , 9, 6, 15. 24 , 42, 35 , 39 , 45

What one of the following is the post-order traversal sequence of the same tree?

a. 6, 15, 9, 24, 21, 39 , 35 , 45 , 27 , 40
b. 6, 15, 9, 24, 21, 35 , 39 , 42 , 45 , 27
c. 6, 15, 9, 24, 21, 39 , 35 , 45 , 42 27
d. 6, 15, 9, 24, 21, 35 , 39 , 45 , 42 27

Q2. The minimum number of nodes in a full binary and a complete binary tree of height 4 are:

a. 8 nodes in full binary tree and, 15 nodes in complete binary tree
b. 9 nodes in full binary tree and, 16 nodes in complete binary tree
c. 15 nodes in full binary tree and, 8 nodes in complete binary tree
d. 16 nodes in full binary tree and, 9 nodes in complete binary tree

Q3. Consider this function:

```
void funA(int i)
{
        if (i >1)
        {
                funA(i / 2);
                printf("*");
        }
}
```

If we call funA(10), how many * will be printed?

a. 4 times
b. 3 times
c. 2 times
d. 5 times

O4. The minimum possible number of nodes of an AVL tree of height 5 is:

a. 20 nodes
b. 12 nodes
c. 19 nodes
d. 22 nodes

Q5. There are 3 different binary trees containing 11, 16, and 13 nodes respectively. Which of those trees cannot be a full binary tree?

a. Tree with 11 nodes
b. Tree with 13 nodes
c. Tree with 16 nodes
d. a and c

Q6. If you have a min heap stored in an array, if the index of the left child is 9, Then the index of the parent of this child will be (Assuming root is at index 0):

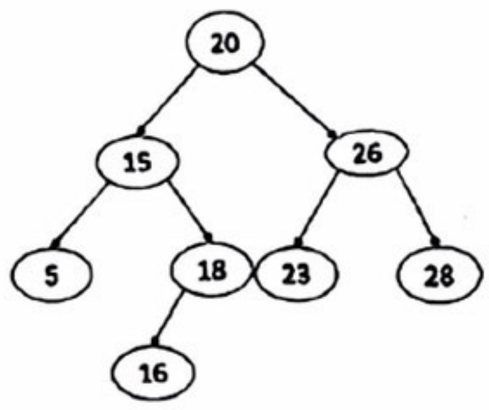
a. 5
b. 4
c. 6
d. 19

Q7. What feature of heaps allows them to be efficiently implemented using an array?

a. Heaps are complete binary trees
b. Heaps are binary search trees
c. Heaps are full binary trees
d. Heaps contain only integer data

Q8. If the tree on the right is binary search tree and we delete the root. Which node can be the new root?

a. Only the node contains data 23
b. Only the node contains data 18
c. Either the node contains data 18 or 23
d. Another node.



Q9. Consider this method:

```
void funcX(int i)
{
        if (i < 0)
        {
                printf("*");
                funcX(-i);
        }
        else if (i < 10){
                printf("%d", i);
        }
        else {
                printf("*");
                funcX(i/2);
        }
}
```

What value of *i* is directly handled by the stopping case (base case)?

a. i >= 0 and i < 10
b. i >= 0 or i < 10
c. i < 10
d. i < 0

Q10. The following numbers are inserted into an empty max heap and an empty binary search tree in the given order: *15 ,20, 5, 25, 17, 30, 19, 12, 35, 28.*

The height of the heap and binary search tree is:

a. Height of the heap is 3 and height of the binary search tree is 3
b. Height of the heap is 4 and height of the binary search tree is 3
c. Height of the heap is 3 and height of the binary search tree is 4
d. Height of the heap is 4 and height of the binary search tree is 4

Q11. If there are N elements stored in a tree representation, in the worst case: the search complexity will be ___ if the tree is a binary tree, and _____ if the tree is an AVL tree.

a. O(N), O(Log N) ,
b. O(N), O(N)
c. O(Log N) , O(Log N) ,
d. O(Log N) , O(N)

Q12. The maximum height of a degree 4 tree containing 8 nodes is:

a. 2
b. 4
c. 6
d. Noe of the above

Q13. Which condition leads to worst-case time complexity in a binary search tree?

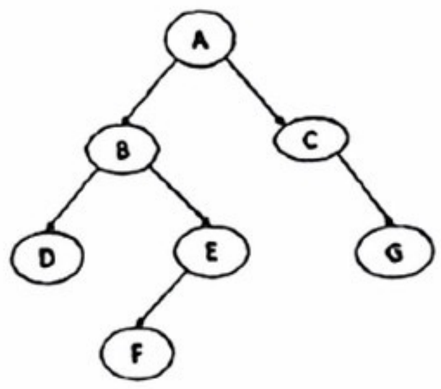
a. Tree is balanced
b. Tree is full
c. Tree is skewed
d. Tree is complete

Q14. The balance factor of node A in an AVL tree was 0 and, a new node was inserted to the left of node A then:

a. Node A may lose its balance
b. Node A will not lose its balance
c. Right child of node A will lose its balance
d. None of the above

Q15. How many leave nodes are in this tree?

a. 6 nodes
b. 3 nodes
c. 4 nodes
d. 7 nodes



Q16. *The sequence of insertion 20,30,25 will lead to imbalance, by using which rotation can be used to balance the tree:*

a. *Left rotation*
b. *Left-Right rotation*
c. *Right-Left rotation*
d. *Right rotation*

Q17. *Choose the correct statement about the min heap*

a. We can insert only at the last level of the tree- as far left a possible.
b. We can delete only the root
c. We can replace the root with the last leaf in the tree only
d. All of above

Q18. Suppose we have numbers between 1 and 100 in a binary search tree and want to search for the number 54. Which of the following sequences could be the sequence of nodes examined?

a. 10, 2, 1, 15, 40, 35, 75, 60, 65
b. 20, 28, 35, 40, 25, 6, 5, 4, 3, 2, 1
c. 80, 60, 55, 10, 20, 25, 30, 40, 50
d. 10, 20, 60, 40, 4, 3, 2, 1, 20, 40

Q19. *In an AVL tree the difference between heights of left and right sub trees cannot be more than:*

a. *0*
b. *1*
c. *2*
d. *3*

Q20. *For Left-Left imbalance which rotation is need to balance the tree?*

a. *Single Left rotation*
b. *Single Right rotation*
c. *Left rotation followed by Right rotation*
d. *Right rotation followed by Left rotation*