1. What is the minimum possible depth of a binary tree with 6 leaves?

A. 2
B. 3
C. 4
D. 5
E. 6

2. What is the minimum possible depth of a binary tree with 10 leaves?

A. 3
B. 4
C. 5
D. 6
E. 7

3. What is the minimum possible depth of a binary tree with 100 leaves?

A. 5
B. 6
C. 7
D. 8
E. 9

4. What is the minimum possible depth of a binary tree with N leaves?

A. $\lceil \log_2(N) \rceil$
B. $\lfloor \log_2(N) \rfloor$
C. $\lceil \log_2(N) \rceil - 1$
D. $\lfloor \log_2(N) \rfloor + 1$

5. What is the minimum possible depth of a ternary tree with 10 leaves?

A. 2
B. 3
C. 4
D. 5

6. What is the minimum possible depth of a ternary tree with 1000 leaves?

A. 5
B. 6
C. 7
D. 8
E. 9

7. What is the minimum possible depth of a ternary tree with N leaves?

A. $\lceil \log_3(N) \rceil$
B. $\lfloor \log_3(N) \rfloor$
C. $\lceil \log_3(N) \rceil - 1$
D. $\lfloor \log_3(N) \rfloor + 1$

8. There are 7 nodes in total in a given regular binary tree. What is the number of its leaves?

A. 2
B. 3
C. 4
D. 5
E. 6

9. There are 11 nodes in total in a given regular binary tree. What is the number of its leaves?

1. 3
2. 4
3. 5
4. 6
5. 7

10. There are 101 nodes in total in a given regular binary tree. What is the number of its leaves?

A. 45
B. 49
C. 50
D. 51
E. 52

11. There are N nodes in total in a given regular binary tree. What is the number of its leaves?

$\log_2(N)$

A. $(N-1) / 2$
B. $N / 2$
C. $(N+1) / 2$
D. $N / 2 + \log_2(N)$

12. A given binary tree T has exactly three leaves. Therefore

a) T has at most two inner nodes,
b) number of inner nodes is not limited,
c) all leaves are in the same depth,
d) all leaves cannot have the same depth,
e) T is regular.

13. Algorithm A traverses a balanced binary tree with n nodes. In each node the algorithm performs an additional procedure which complexity is $\Theta(1)$. What is the asymptotic complexity of A?

A. $\Theta(1)$
B. $\Theta(\log_2(n))$
C. $\Theta(n)$
D. $\Theta(n \times \log_2(n))$
E. $\Theta(n^2)$

14. Algorithm A traverses a balanced binary tree with n nodes. In each node the algorithm performs an additional procedure which complexity is $\Theta(n)$. What is the asymptotic complexity of A?

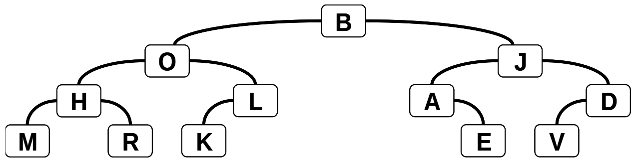
A. $\Theta(1)$
B. $\Theta(\log_2(n))$
C. $\Theta(n)$
D. $\Theta(n \times \log_2(n))$
E. $\Theta(n^2)$

15. Algorithm A traverses a balanced binary tree with n nodes. In each node the algorithm performs an additional procedure which complexity is $\Theta(n^2)$. What is the asymptotic complexity of A?

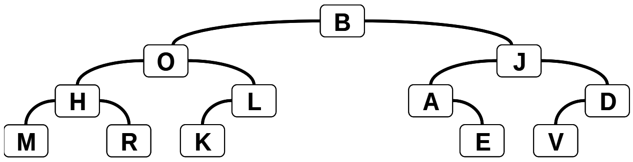
16. Algorithm A traverses a binary tree with depth D. The total number of operations performed in depth k is equal to *D*. Each operation has a constant asymptotic complexity. What is the asymptotic complexity of A?

17. Algorithm A traverses a binary tree with depth D. The total number of operations performed in depth k is equal to $k+D$. Each operation has a constant asymptotic complexity. What is the asymptotic complexity of A?

18. Algorithm A traverses the given tree and in each node it prints the character stored in that node. Write the output of the algorithm when the traversal is
-- Inorder



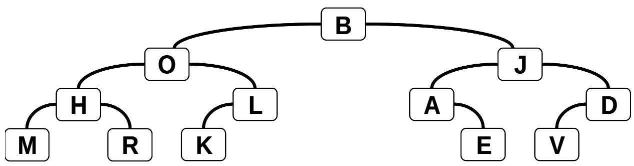
19. Algorithm A traverses the given tree and in each node it prints the character stored in that node. Write the output of the algorithm when the traversal is



-- Preorder

20. Algorithm A traverses the given tree and in each node it prints the character stored in that node. Write the output of the algorithm when the traversal is

-- Postorder



21. A method first prints the keys of the nodes of a binary tree using the Inorder traversal an then it prints the keys of the nodes using the Preorder traversal. The result is
Inorder: 45 71 98 47 50 62 87 3 79
Preorder: 50 47 71 45 98 62 3 87 79

Reconstruct the tree completely.

22. Write an algorithm which will reconstruct any binary tree when it is given a sequence S1 of keys of the nodes produced by the Inorder traversal and a sequence S2 of keys of the nodes produced by the Preorder traversal.

23. We have to traverse a regular binary tree and visit all its n nodes. We can move along each edge only in the direction from the root towards a leaf. We can also jump from any node directly back to the root. Each move along one edge and each jump to the root takes one millisecond. Compute the complexity of the traversal. The tree is perfectly balanced.

24. We have to traverse a regular binary tree and visit all its n nodes. We can move along each edge only in the direction from the root towards a leaf. We can also jump from any node directly back to the root. Each move along one edge and each jump to the root takes one millisecond. Compute the complexity of the traversal. The tree is maximally disbalanced (its depth is $(n-1)/2$).

25. Write a function which will remove (delete) all the leaves of a given tree. Suppose there is a reference to the parent in each node. Can the same task be achieved in a tree where nodes do not contain a reference to the parent? What is the complexity of your solution?

26. The height of a node X is defined as the number of edges on a path from X to the most distant leaf in the subtree which root is X. Write the function which will assign each node in the tree the value of its height.

27. Write a function which will create an exact copy of a binary tree.

28. Write a function which modify a binary tree in such way that it will become a mirror copy of itself. This means that the Inoder traversal of the original tree and the same inorder traversal of the modified tree will yield sequences which are reverse of each other.

29. Write a function which will merge two binary trees in the following way:
It removes one of the deepest leaves in the tree which depth is not smaller the the depth of the other tree.

30. Write a function which will merge two binary trees in the following way:
The removed node becomes the root of the new tree. The tree from which the node was removed becomes the left subtree of the root and the other tree becomes the right subtree of the root.
No nodes should be physically deleted by the operation, you should only manipulate the references of the nodes.

31. An arithmetic expression containing only positive integers, brackets and symbols of operations +,–,*,/ can be represented as a binary tree.
Example:The expression 6 + (4-3+5)*(9-7) is represented by the tree in the picture.

Write a node representation and a function which input is the reference to the tree root and the return value is equal to the value of the expression represented by the tree.