Elementary Data Structures and Algorithms

Binary and binary search trees

Concept: *complexity*

1. Consider a worst-case binary search tree with *n* nodes. What is the average case time complexity for finding a value at a leaf?

   A. log *n*

   B. quadratic

   C. linear

   D. n

   E. constant

   F. *n* log *n*

2. Consider a binary search tree with *n* nodes. What is the worst case time complexity for finding a value at a leaf?

   A. *n* log *n*

   B. quadratic

   C. n

   D. constant

   E. log *n*

   F. linear

3. Consider a binary search tree with *n* nodes. What is the minimum and maximum height (using order notation)?

   A. linear and linear

   B. constant and linear

   C. constant and log *n*

   D. log *n* and log *n*

   E. log *n* and linear

Concept: *balance*

4. Which ordering of input values builds the most unbalanced BST? Assume values are inserted from left to right.

   A. 4 3 1 6 2 8 7

   B. 1 2 3 4 5 7 6

   C. 1 7 2 6 3 5 4

5. Which ordering of input values builds the most balanced BST? Assume values are inserted from left to right.

   A. 1 4 3 2 5 7 6

   B. 1 2 7 6 0 3 8

C. 4 3 1 6 2 8 7

Concept: *tree shapes*

6. What is the best definition of a perfect binary tree?

   A. all leaves are equidistant from the root

   B. all null children are equidistant from the root

   C. all nodes have zero or two children

   D. all leaves have zero children

7. Suppose a binary tree has 10 leaves. How many nodes in the tree must have two children?

   A. 8

   B. no limit

   C. 9

   D. 7

   E. 10

8. Suppose a binary tree has 10 nodes. How many nodes are children of some other node in the tree?

   A. 7

   B. 9

   C. 10

   D. 8

   E. no limit

9. Let P0, P1, and P2 refer to nodes that have zero, one or two children, respectively. Using the generally accepted definition, what is a *full* binary tree?

   A. all interior nodes P1, except the root

   B. all interior nodes are P1

   C. all nodes are P2

   D. all interior nodes are P2

   E. all interior nodes are P2; all leaves are equidistant from the root

   F. all leaves are equidistant from the root

10. Let P0, P1, and P2 refer to nodes that have zero, one or two children, respectively. Using the generally accepted definition, what is a *perfect* binary tree?

    A. all interior nodes are P2

    B. all interior nodes P1, except the root

    C. all interior nodes are P1

    D. all nodes are P0 or P2

    E. all interior nodes are P2; all leaves are equidistant from the root

    F. all leaves are equidistant from the root

11. Let P0, P1, and P2 refer to nodes that have zero, one or two children, respectively. Using the generally accepted definition, what is a *degenerate* binary tree?

    A. all interior nodes are P2

    B. all interior nodes are P1

    C. all leaves are equidistant from the root

    D. all interior nodes are P2; all leaves are equidistant from the root

    E. all interior nodes P1, except the root

    F. all nodes are P0 or P2

12. Let P0, P1, and P2 refer to nodes that have zero, one or two children, respectively. Using the generally accepted definition of a *complete* binary tree, which of the following actions can be used to make any complete tree? Assume the leftmost and rightmost sets may be empty.

    A. making the leftmost leaf of a *perfect* tree P1

    B. another name for a *perfect* tree

    C. removing the rightmost leaves from a *perfect* tree

    D. making the leftmost leaves of a *perfect* tree P1 or P2

    E. making the leftmost leaves of a *perfect* tree P2

13. **T** or **F**: All *perfect* trees are *full* trees.

14. **T** or **F**: All *full* trees are *complete* trees.

15. **T** or **F**: All *complete* trees are *perfect* trees.

16. How many distinct binary search trees can be formed from exactly two nodes with values 1, 2, or 3 respectively (hint: think about how many permutations of values there are for each tree shape)?

    A. 6

    B. 3

    C. 2

    D. 4

    E. 5

17. How many distinct binary tree shapes can be formed from exactly two nodes?

    A. 2

    B. 1

    C. 3

    D. 5

    E. 4

18. Let *k* be the the number of steps from the root to a leaf in a perfect tree. What are the number of nodes in the tree?

    A. 2 k + 1

    B. 2 k - 1 - 1

    C. 2 k + 1 - 1

    D. 2 k - 1 + 1

E. 2 k - 1

19. Let *k* be the the number of steps from the root to the furthest leaf in a binary tree. What would be the minimum number of nodes in such a tree? Assume *k* is a power of two.

    A. log *k*

    B. 2 k + 1 - 1

    C. (log *k*) + 1

    D. 2 k + 1

    E. *k* + 1

    F. *k*

20. Let *k* be the the number of steps from the root to the furthest leaf in a binary tree. What would be the maximum number of nodes in such a tree? Assume *k* is a power of two.

    A. log *k*

    B. *k* + 1

    C. *k*

    D. 2 k + 1 - 1

    E. (log *k*) + 1

    F. 2 k + 1

Concept: *ordering in a BST*

21. For all child nodes in a BST, what relationship holds between the value of a left child node and the value of its parent? Assume unique values.

    A. there is no relationship

    B. greater than

    C. less than

22. For all sibling nodes in a BST, what relationship holds between the value of a left child node and the value of its sibling? Assume unique values.

    A. there is no relationship

    B. less than

    C. greater than

23. Which statement is true about the *successor* of a node in a BST, if it exists?

    A. has no left child

    B. has no right child

    C. it is always a leaf node

    D. it may be an ancestor

    E. it is always an interior node

24. Consider a node which holds neither the smallest or the largest value in a BST. Which statement is true about the node which holds the next higher value of a node in a BST, if it exists?

    A. it is always an interior node

    B. it may be an ancestor

C. it is always a leaf node

D. has no right child

E. has no left child

Concept: *traversals*

25. Consider printing out the node values of a binary tree with 25 nodes to the left of the root and 38 nodes to the right. How many nodes are processed before the root's value is printed in a pre-order traversal?

    A. 38

    B. none of the other answers are correct

    C. 53

    D. 54

    E. 25

    F. 0

26. Consider printing out the node values of a binary tree with 25 nodes to the left of the root and 38 nodes to the right. How many nodes are processed before the root's value is printed in an in-order traversal?

    A. 25

    B. 54

    C. 38

    D. 53

    E. none of the other answers are correct

    F. 0

27. Consider printing out the node values of a binary tree with 25 nodes to the left of the root and 38 nodes to the right. How many nodes are processed before the root's value is printed in a post-order traversal?

    A. 0

    B. 63

    C. 25

    D. none of the other answers are correct

    E. 54

    F. 38

28. Consider a perfect BST with even values 0 through 12, to which the value 7 is then added. Which of the following is an in-order traversal of the resulting tree?

    A. 7 0 2 4 6 8 10 12

    B. 0 4 2 7 8 10 12 6

    C. 6 2 10 0 4 8 12 7

    D. 0 2 4 6 7 8 10 12

    E. 0 2 4 6 8 10 12 7

    F. 12 10 8 7 6 4 2 0

29. Consider a perfect BST with even values 0 through 12, to which the value 7 is then added. Which of the following is a level-order traversal of the resulting tree?

A. 6 2 10 0 4 8 12 7

B. 0 4 2 7 8 10 12 6

C. 0 2 4 6 7 8 10 12

D. 12 10 8 7 6 4 2 0

E. 7 0 2 4 6 8 10 12

F. 0 2 4 6 8 10 12 7

30. Consider a level-order traversal of C B A D F E and an in-order traversal of B C A F D E . Do these traversals generate a unique tree and, if so, what is that tree's pre-order traversal?

    A. yes, C A D B E F

    B. yes, but the correct answer is not listed

    C. yes, C B A D F E

    D. yes, C B A F D E

    E. no

31. Consider an in-order traversal of B C A F D E and a pre-order traversal of C B A D F E . Do these traversals generate a unique tree and, if so, what is that tree's post-order traversal?

    A. yes, B F A E D C

    B. yes, but the correct answer is not listed

    C. yes, B F E D A C

    D. yes, F A D B E C

    E. no

32. Consider an in-order traversal of B C A F D E and a post-order traversal of C B A D F E . Do these traversals generate a unique tree and, if so, what is that tree's level-order traversal?

    A. no

    B. yes, E A C F B D

    C. yes, E F C D B A

    D. yes, E F A D B C

    E. yes, but the correct answer is not listed

33. Consider a level-order traversal of C F D E B A and an pre-order traversal of C F E A D B . Do these traversals generate a unique tree and, if so, what is that tree's in-order traversal?

    A. yes, but the correct answer is not listed

    B. yes, F A E C B D

    C. yes, F E A C D B

    D. yes, F A E C B D

    E. no

Concept: *insertion and deletion*

34. **T** or **F**: Suppose you are given an in-order traversal of an unbalanced BST. If you were to insert those values into an empty BST in the order given, would the result be a balanced tree?

35. **T** or **F**: Suppose you are given a pre-order traversal of an unbalanced BST. If you were to insert those values into an empty BST in the order given, would the result be a balanced tree?

36. **T** or **F**: Suppose you are given an in-order traversal of a balanced BST. If you were to insert those values into an empty BST in the order given, would the result be a balanced tree?

37. **T** or **F**: Suppose you are given a pre-order traversal of a balanced BST. If you were to insert those values into an empty BST in the order given, would the result be a balanced tree?

38. Suppose 10 values are inserted inserted into an empty BST. What is the minimum and maximum resulting heights of the tree? The height is the number of steps from the root to the furthest leaf.

    A. 3 and 10

    B. 3 and 9

    C. 4 and 9

    D. 5 and 10

    E. 4 and 10

    F. 5 and 9

39. Which, if any, of these deletion strategies for non-leaf nodes reliably preserve BST ordering?

    i. Swap the values of the node to be deleted and the smallest leaf node with a larger value, then remove the leaf.

    ii. Swap the values of the node to be deleted with its predecessor or successor. If the predecessor or successor is a leaf, remove it. Otherwise, repeat the process.

    iii. If the node to be deleted does not have two children, simply connect the parent's child pointer to the node to the node's child pointer, otherwise, use a correct deletion strategy for nodes with two children.

    A. *ii*

    B. *iii*

    C. none

    D. *ii* and *iii*

    E. *i* and *iii*

    F. *i*

    G. all

    H. *i* and *ii*