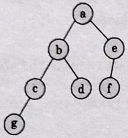
5 Trees

5.1 In the balanced binary tree in the figure given below, how many nodes will become unbalanced when a node is inserted as a child of the node "g"?



(a) 1 (b) 3
(c) 7 (d) 8 **[1996 : 1 M]**

5.2 Which of the following sequences denotes the post order traversal sequence of the tree of above question?
(a) fegcbdba (b) gcbdafe
(c) gcdbfea (d) fedgcba **[1996 : 1 M]**

5.3 A binary search tree is generated by inserting in order of the following integers: 50, 15, 62, 5, 20, 58, 91, 3, 8, 37, 60, 24. The number of nodes in the left subtree and right subtree of the root respectively is
(a) (4, 7) (b) (7, 4)
(c) (8, 3) (d) (3, 8) **[1996 : 2 M]**

5.4 A binary search tree is used to locate the number 43. Which of the following probe sequences are possible and which are not? Explain.
(a) 61 52 14 17 40 43 (b) 2 3 50 40 60 43
(c) 10 65 31 48 37 43 (d) 81 61 52 14 41 43
(e) 17 77 27 66 18 43 **[1996 : 2 M]**

5.5 A binary search tree is generated by inserting in order the following integers: 50, 15, 62, 5, 20, 58, 91, 3, 8, 37, 60, 24. The number of nodes in the left subtree and right subtree of the root respectively is
(a) (4, 7) (b) (7, 4)
(c) (8, 3) (d) (3. 8) **[1996 : 2 M]**

5.6 A binary search tree contains the values 1, 2, 3, 4, 5, 6, 7, 8. The tree is traversed in pre-order and the values are printed out. Which of the following sequences is a valid output?
(a) 53124786 (b) 53126487
(c) 53241678 (d) None of these
[1997 : 2 M]

5.7 Which of the following statements is false?
(a) A tree with n nodes has (n – 1) edges
(b) A labeled rooted binary tree can be uniquely constructed given its postorder and preorder traversal results.
(c) A complete binary tree with n internal nodes has (n + 1) leaves
(d) The maximum number of nodes in a binary tree of height h is $(2^{h+1} - 1)$
[1998 : 1 M]

5.8 A complete n-ary tree is one in which every node has 0 or n sons. If x is the number of internal nodes of a complete n-ary tree, the number of leaves in it is given by
(a) x (n – 1) + 1 (b) xn – 1
(c) xn + 1 (d) x(n + 1)
[1998 : 2 M]

5.9 Consider the following nested representation of binary trees: (X Y Z) indicates Y and Z are the left and right subtrees, respectively, of node X. Note that Y and Z may be NULL, or further nested. Which of the following represents a valid binary tree?
(a) (1 2 (4 5 6 7))
(b) (1 ((234) 56)7)
(c) (1(234) (567))
(d) (1(23NULL) (45))
[2000 : 1 M]

5.10 Let LASTPOST, LASTIN and LASTPRE denote the last vertex visited in a postorder, inorder and preorder traversal. Respectively, of a complete binary tree. Which of the following is always true?
(a) LASTIN = LASTPOST
(b) LASTIN = LASTPRE
(c) LASTPRE = LASTPOST
(d) None of the above **[2000 : 2 M]**

5.11 The number of leaf nodes in a rooted tree of n nodes, with each node having 0 or 3 children is
(a) $\frac{n}{2}$ (b) $\frac{(n-1)}{3}$
(c) $\frac{(n-1)}{2}$ (d) $\frac{(2n+1)}{3}$
[2002 : 2 M]

5.12 Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty binary search tree. The binary search tree uses the usual ordering on natural numbers. What is the inorder traversal sequence of the resultant tree?
(a) 7 5 1 0 3 2 4 6 8 9
(b) 0 2 4 3 1 6 5 9 8 7
(c) 0 1 2 3 4 5 6 7 8 9
(d) 9 8 6 4 2 3 0 1 5 7 **[2003 : 1 M]**

5.13 A data structure is required for storing a set of integers such that each of the following operations can be done in O(log n) time, where n is the number of elements in the set.
1. Deletion of the smallest element.
2. Insertion of an element if it is not already present in the set.

Which of the following data structures can be used for this purpose?
(a) A heap can be used but not a balanced binary search tree
(b) A balanced binary search tree can be used but not a heap
(c) Both balanced binary search tree and heap can be used
(d) Neither balanced binary search tree nor heap can be used
[2003 : 2 M]

5.14 The following numbers are inserted into an empty binary search tree in the given order: 10, 1, 3, 5, 15, 12, 16. What is the height of the binary search tree (the height is the maximum distance of a leaf node from the root)?
(a) 2 (b) 3
(c) 4 (d) 6 **[2004 : 1 M]**

5.15 Level order traversal of a rooted tree can be done by starting from the root and performing
(a) preorder traversal
(b) inorder traversal
(c) depth first search
(d) breadth first search **[2004 : 1 M]**

5.16 Consider the following C program segment:

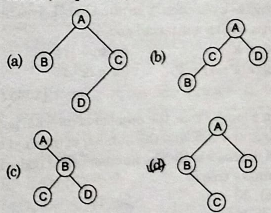
```
struct CellNode
{
  struct CellNode *leftChild;
  int element;
  struct CellNode *rightChild;
};
int DoSomething (struct CellNode *ptr)
{
  int value = 0;
  if (ptr ! = NULL)
  {
    if (ptr → leftChild ! = NULL)
      value = 1 + DoSomething (ptr → leftChild);
    if (ptr → rightChild ! = NULL)
      value = max (value, 1 + DoSomething (ptr
      → rightChild));
  }
  return (value);
}
```

The value returned by the function DoSomething when a pointer to the root of a non-empty tree is passed as argument is
(a) The number of leaf nodes in the tree
(b) The number of nodes in the tree
(c) The number of internal nodes in the tree
(d) The height of the tree **[2004 : 2 M]**

5.17 Consider the label sequences obtained by the following pairs of traversals on a labeled binary tree. Which of these pairs identify a tree uniquely?
1. Preorder and postorder
2. Inorder and postorder
3. Preorder and inorder
4. Level order and postorder

(a) 1 only (b) 2 and 3
(c) 3 only (d) 4 only **[2004 : 2 M]**

5.18 Which one of the following binary trees has its inorder and preorder traversals as BCAD and ABCD, respectively?



[2004 : 2 M]

5.19 The numbers 1, 2,..., n are inserted in a binary search tree in some order. In the resulting tree, the right subtree of the root contains p nodes. The first number to be inserted in the tree must be
(a) p (b) p + 1
(c) n – p (d) n – p + 1
[2005 : 1 M]

5.20 In a binary tree, for every node the difference between the number of nodes in the left and right subtrees is at most 2. If the height of the tree is h > 0, then the minimum number of nodes in the tree is

(a) 2^{h-1} (b) $2^{h-1}+1$
(c) $2^h - 1$ (d) 2^h **[2005 : 2 M]**

5.21 A binary search tree contains the numbers 1, 2, 3, 4, 5, 6, 7, 8. When the tree is traversed in preorder and the values in each node printed out, the sequence of values obtained is 5, 3, 1, 2, 4, 6, 8, 7. If the tree is traversed in postorder, the sequence obtained would be

(a) 8, 7, 6, 5, 4, 3, 2, 1
(b) 1, 2, 3, 4, 8, 7, 6, 5
(c) 2, 1, 4, 3, 6, 7, 8, 5
(d) 2, 1, 4, 3, 7, 8, 6, 5 **[2005 : 2 M]**

5.22 Postorder traversal of a given binary search tree, T produces the following sequence of keys
10, 9, 23, 22, 27, 25, 15, 50, 95, 60, 40, 29
Which one of the following sequences of keys can be the result of an inorder traversal of the tree T?

(a) 9, 10, 15, 22, 23, 25, 27, 29, 40, 50, 60, 95
(b) 9, 10, 15, 22, 40, 50, 60, 95, 23, 25, 27, 29
(c) 29, 15, 9, 10, 25, 22, 23, 27, 40, 60, 50, 95
(d) 95, 50, 60, 40, 27, 23, 22, 25, 10, 0, 15, 29 **[2005 : 2 M]**

5.23 In a complete k-ary, every internal node has exactly k children. The number of leaves in such a tree with n internal nodes is

(a) n k (b) (n – 1) k +1
(c) n(k – 1) + 1 (d) n (k –1) **[2005 : 2 M]**

5.24 In a binary tree, the number of internal nodes of degree 1 is 5, and the number of internal nodes of degree 2 is 10. The number of leaf nodes in the binary tree is

(a) 10 (b) 11
(c) 12 (d) 15 **[2006 : 1 M]**

5.25 Suppose that we have numbers between 1 and 100 in a binary search tree and want to search for the number 55. Which of the following sequences CANNOT be the sequence of nodes examined?

(a) {10, 75, 64, 43, 60, 57, 55}
(b) {90, 12, 68, 34, 62, 45, 55}
(c) {9, 85, 47, 68, 43, 57, 55}
(d) {79, 14, 72, 56, 16, 53, 55} **[2006 : 2 M]**

Directions for Question 5.26 to 5.28:

An array X of n distinct integers is interpreted as a complete binary tree. The index of the first element of the array is 0.

5.26 The index of the parent of element X [i], $i \neq 0$, is

(a) $\left\lfloor \frac{i}{2} \right\rfloor$ (b) $\left\lceil \frac{(i-1)}{2} \right\rceil$
(c) $\left\lceil \frac{i}{2} \right\rceil$ (d) $\left\lceil \frac{i}{2} \right\rceil - 1$ **[2006 : 2 M]**

5.27 If only the root node does not satisfy the heap property, the algorithm to convert the complete binary tree into a heap has the best asymptotic time complexity of

(a) O(n)
(b) O(log n)
(c) O(n log n)
(d) O(n loglog n) **[2006 : 2 M]**

5.28 If the root node is at level 0, the level of element X[i], $i \neq 0$, is

(a) $\lfloor \log_2 i \rfloor$ (b) $\lceil \log_2 (i+1) \rceil$
(c) $\lfloor \log_2 (i+1) \rfloor$ (d) $\lceil \log_2 i \rceil$ **[2006 : 2 M]**

5.29 The height of a binary tree is the maximum number of edges in any root to leaf path. The maximum number of nodes is a binary tree of height h is

(a) 2^h (b) $2^{h-1}-1$
(c) $2^{h+1}-1$ (d) 2^{h+1} **[2007 : 1 M]**

5.30 The maximum number of binary trees that can be formed with three unlabeled nodes is

(a) 1 (b) 5
(c) 4 (d) 3 **[2007 : 1 M]**

5.31 The inorder and preorder traversal of a binary tree are *dbeafcg* and *abdecfg* respectively. The postorder traversal of the binary tree is

(a) debfgca (b) edbgfca
(c) edbfgca (d) defgbca **[2007 : 2 M]**

5.32 A complete n-ary tree is a tree in which each node has n children or no children. Let *I* be the number of internal nodes and L be the number of leaves in a complete n-ary tree. If L = 41, and *I* = 10, what is the value of n

(a) 3 (b) 4
(c) 5 (d) 6 **[2007 : 2 M]**

5.33 Consider the following C program segment where CellNode represents a node in a binary tree

```
struct CellNode {
struct CelloNode * leftchild;
int element;
struct CellNode *rightchild;
};
int GetValue (struct CellNode * ptr) {
    int value = 0;
if (ptr! = NULL) {
    if ((ptr → leftChild = = NULL) &&
        (ptr → rightChild = = NULL))
        Value = 1;
else
    value = value +
        GetValue (ptr→leftChild)
        + GetValue(ptr→ rightChild);
}
    return (value);
}
```

The value returned by Get Value when a pointer to the root of a binary tree is passed as its argument is

(a) the number of nodes
(b) the number of internal nodes in the tree
(c) the number of leaf nodes in the tree
(d) the height of the tree **[2007 : 2 M]**

5.34 When searching for the key value 60 in a binary search tree, nodes containing the key values 10, 20, 40, 50, 70 80, 90 are traversed, not necessarily in the order given. How many different orders are possible in which these key values can occur on the search path from the root to the node containing the value 60?

(a) 35 (b) 64
(c) 128 (d) 5040 **[2007 : 2 M]**

5.35 Which of the following is TRUE?

(a) The cost of searching an AVL tree is Θ(log n) but that of a binary search tree is O(n)
(b) The cost of searching an AVL tree is Θ(log n) but that of a complete binary tree is Θ(n log n)
(c) The cost of searching a binary search tree is O (log n) but that of an AVL tree is Θ(n)
(d) The cost of searching an AVL tree is Θ(n log n) but that of a binary search tree is O(n) **[2008 : 1 M]**

Linked Questions 5.36 & 5.37:

A binary tree with n > 1 nodes has n_1, n_2 and n_3 nodes of degree one, two and three respectively. The degree of a node is defined as the number of its neighbours.

5.36 n_3 can be expressed as:

(a) $n_1 + n_2 - 1$ (b) $n_1 - 2$
(c) $\left\lceil \left(\frac{(n_1 + n_2)}{2} \right) \right\rceil$ (d) $n_2 - 1$ **[2008 : 2 M]**

5.37 Starting with the above tree, while there remains a node v of degree two in the tree, add an edge between the two neighbours of v and then remove v from the tree.
How many edges will remain at the end of the process?

(a) $2 * n_1 - 3$
(b) $n_2 + 2 * n_1 - 2$
(c) $n_3 - n_2$
(d) $n_2 + n_1 - 2$ **[2008 : 2 M]**

5.38 You are given the postorder traversal, P of a binary search tree on the n elements 1, 2, ….. , n. You have to determine the unique binary search tree that has P as its postorder traversal. What is the time complexity of the most efficient algorithm for doing this?

(a) Θ(log n)
(b) Θ(n)
(c) Θ(n log n)
(d) none of the above, as the tree cannot be uniquely determined. **[2008 : 2 M]**

5.39 We have a binary heap on n elements and wish to insert n more elements (not necessarily one after another) into this heap. The total time required for this is

(a) Θ(log n) (b) Θ(n)
(c) Θ(n log n) (d) $\Theta(n^2)$ **[2008 : 2 M]**

5.40 The following three are known to be the preorder, inorder and postorder sequences of a binary tree. But it is not known which is which.

I. MBCAFHPYK
II. KAMCBYPFH
III. MABCKYFPH

Pick the true statement from the following.

(a) I and II are preorder and inorder sequences, respectively
(b) I and III are preorder and postorder sequences, respectively
(c) II is the inorder sequence, but nothing more can be said about the other two sequences
(d) II and III are the preorder and inorder sequences, respectively **[2008 : 2 M]**

5.41 What is the maximum height of any AVL-tree with 7 nodes? Assume that the height of a tree with a single node is 0.

(a) 2 (b) 3
(c) 4 (d) 5 **[2009 : 2 M]**

5.42 The height of a tree is defined as the number of edges on the longest path in the tree. The function shown in the pseudocode below is invoked as the height (root) to compute the height of a binary tree rooted at the tree pointer root.

```
int height(treeptr n)  {
   if(n==NULL)return –1;
   if(n → left==NULL)
      if(n→right==NULL)return0;
      else return [B1] ;//Box1
else { h1=height(n → left);
      if(n → right==NULL) return(1+h1);
      else {h2=height(n → right);
            return [B2] ;//Box2 }
   }
}
```

The appropriate expressions for the two boxes B1 and B2 are

(a) B1: (1 + height (n → right))
B2: (1 + max(h1, h2))
(b) B1: (height(n → right))
B2: (1 + max(h1, h2))
(c) B1: height(n → right)
B2: max(h1, h2)
(d) B1: (1 + height(n → right))
B2: max(h1, h2)

[2012 : 2 M]

5.43 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?

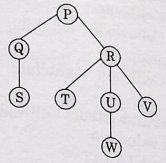
(a) 10, 20, 15, 23, 25, 35, 42, 39, 30
(b) 15, 10, 25, 23, 20, 42, 35, 39, 30
(c) 15, 20, 10, 23, 25, 42, 35, 39, 30
(d) 15, 10, 23, 25, 20, 35, 42, 39, 30

[2013 : 2 M]

5.44 Consider a rooted n node binary tree represented using pointers. The best upper bound on the time required to determine the number of subtrees having exactly 4 nodes is $O(n^a \log^b n)$. Then the value of $a + 10b$ is ____1____.

[2014 (Set-1) : 1 M]

5.45 Consider the following rooted tree with the vertex labeled P as the root:



The order in which the nodes are visited during an in-order traversal of the tree is

(a) SQPTRWUV
(b) SQPTUWRV
(c) SQPTWUVR
(d) SQPTRUWV

[2014 (Set-3) : 1 M]

5.46 Consider the pseudocode given below. The function **DoSomething()** takes as argument a pointer to the root of an arbitrary tree represented by the *leftMostChild-rightSibling* representation. Each node of the tree is of type **treeNode**.

```
typedef struct treeNode* treeptr;
struct treeNode
{
    treeptr leftMostChild, rightSibling;
};
int DoSomething (treeptr tree)
{
    int value=0;
    if (tree != NULL)
    {
        if (tree->leftMostChild == NULL)
            value = 1;
        else
            value = DoSomething(tree->
                                leftMostChild);
        value=value+DoSomething(tree->
                                rightSibling);
    }
    return(value);
}
```

When the pointer to the root of a tree is passed as the argument to **DoSomething**, the value returned by the function corresponds to the

(a) number of internal nodes in the tree.
(b) height of the tree.
(c) number of nodes without a right sibling in the tree.
(d) number of leaf nodes in the tree.

[2014 (Set-3) : 2 M]

5.47 Which of the following is/are correct inorder traversal sequence(s) of binary search tree(s)?

1. 3, 5, 7, 8, 15, 19, 25
2. 5, 8, 9, 12, 10, 15, 25
3. 2, 7, 10, 8, 14, 16, 20
4. 4, 6, 7, 9, 18, 20, 25

(a) 1 and 4 only (b) 2 and 3 only
(c) 2 and 4 only (d) 2 only

[2015 (Set-1) : 1 M]

5.48 The height of a tree is the length of the longest root-to-leaf path in it. The maximum and minimum number of nodes in a binary tree of height 5 are

(a) 63 and 6, respectively
(b) 64 and 5, respectively
(c) 32 and 6, respectively
(d) 31 and 5, respectively **[2015 (Set-1) : 1 M]**

5.49 A binary tree T has 20 leaves. The number of nodes in T having two children is ____19____.

[2015 (Set-2) : 1 M]

5.50 While inserting the elements 71, 65, 84, 69, 67, 83 in an empty binary search tree (BST) in the sequence shown, the element in the lowest level is

(a) 65 (b) 67
(c) 69 (d) 83

[2015 (Set-3) : 1 M]

5.51 The number of ways in which the numbers 1, 2, 3, 4, 5, 6, 7 can be inserted in an empty binary search tree, such that the resulting tree has height 6, is ____64____. **Note:** The height of a tree with a single node is 0.

[2016 (Set-2) : 2 M]

5.52 Let T be a binary search tree with 15 nodes. The minimum and maximum possible heights of T are:

Note: The height of a tree with a single node is 0.

(a) 4 and 15 respectively
(b) 3 and 14 respectively
(c) 4 and 14 respectively
(d) 3 and 15 respectively **[2017 (Set-1) : 1 M]**

5.53 The pre-order traversal of a binary search tree is given by 12, 8, 6, 2, 7, 9, 10, 16, 15, 19, 17, 20. Then the post-order traversal of this tree is:

(a) 2, 6, 7, 8, 9, 10, 12, 15, 16, 17, 19, 20
(b) 2, 7, 6, 10, 9, 8, 15, 17, 20, 19, 16, 12
(c) 7, 2, 6, 8, 9, 10, 20, 17, 19, 15, 16, 12
(d) 7, 6, 2, 10, 9, 8, 15, 16, 17, 20, 19, 12

[2017 (Set-2) : 2 M]

5.54 The postorder traversal of a binary tree is 8, 9, 6, 7, 4, 5, 2, 3, 1. The inorder traversal of the same tree is 8, 6, 9, 4, 7, 2, 5, 1, 3. The height of a tree is the length of the longest path from the root to any leaf. The height of the binary tree above is ____4____.

[2018 : 1 M]

5.55 Let T be a full binary tree with 8 leaves. (A full binary tree has every level full.) Suppose two leaves a and b of T are chosen uniformly and independently at random. The expected value of the distance between a and b in T (i.e., the number of edges in the unique path between a and b) is (rounded off to 2 decimal places) ____4.25____.

[2019 : 2 M]

5.56 The preorder traversal of a binary search tree is 15, 10. 12, 11, 20, 18, 16, 19.

Which one of the following is the postorder traversal of the tree?

(a) 20, 19, 18, 16, 15, 12, 11, 10
(b) 10, 11, 12, 15, 16, 18, 19, 20
(c) 19, 16, 18, 20, 11, 12, 10, 15
(d) 11, 12, 10, 16, 19, 18, 20, 15 **[2020 : 1 M]**

5.57 Consider the following statements:

S_1: The sequence of procedure calls corresponds to a preorder traversal of the activation tree.

S_2: The sequence of procedure returns corresponds to a postorder traversal of the activation tree.

Which one of the following options is correct?

(a) S_1 is false and S_2 is false
(b) S_2 is true and S_2 is false
(c) S_1 is true and S_2 is true
(d) S_1 is false and S_2 is true

[2021 (Set-1) : 1 M]

5.58 A binary search tree T contains n distinct elements. What is the time complexity of picking an element in T that is smaller than the maximum element in T?

(a) $\theta(n)$ (b) $\theta(n \log n)$
(c) $\theta(\log n)$ (d) $\theta(1)$

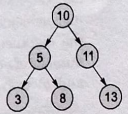
[2021 (Set-1) : 1 M]

5.59 Consider a complete binary tree with 7 nodes. Let A denote the set of first 3 elements obtained by performing Breadth-First Search (BFS) starting from the root. Let B denote the set of first 3 elements obtained by performing Depth-First Search (DFS) starting from the root. The value of $|A - B|$ is ____1____.

[2021 (Set-2) : 1 M]

5.60 Suppose a binary search tree with 1000 distinct elements is also a complete binary tree. The tree is stored using the array representation of binary heap trees. Assuming that the array indices start with 0, the 3rd largest element of the tree is stored at index 509. **[2022 : 1 M]**

5.61 Consider the C function foo and the binary tree shown:




```
typedef struct node {
    int val;
    struct node *left, *right;
} node;
int foo(node *p) {
    int retval;
    if (p == NULL)
        return 0;
    else {
        retval = p->val + foo(p->left) + foo(p->right);
        printf("%d ", retval);
        return retval;
    }
}
```

When foo is called with a pointer to the root node of the given binary tree, what will it print?

(a) 3 8 5 13 11 10 (b) 3 5 8 10 11 13
(c) 3 8 16 13 24 50 (d) 3 16 8 50 24 13

[2023 : 2 M]

5.62 An array [82, 101, 90, 11, 111, 75, 33, 131, 44, 93] is heapified. Which one of the following options represents the first three elements in the heapified array?

(a) 131, 111, 90 (b) 131, 11, 93
(c) 82, 11, 93 (d) 82, 90, 101

[2024 (Set-1) : 2 M]

5.63 Consider a binary min-heap containing 105 distinct elements. Let k be the index (in the underlying array) of the maximum element stored in the heap. The number of possible values of k is

(a) 52 (b) 1
(c) 53 (d) 27

[2024 (Set-1) : 2 M]

5.64 You are given a set V of distinct integers. A binary search tree T is created by inserting all elements of V one by one, starting with an empty tree. The tree T follows the convention that, at each node, all values stored in the left subtree of the node are smaller than the value stored at the node. You are not aware of the sequence in which these values were inserted into T, and you do not have access to T.

Which one of the following statements is TRUE?

(a) Preorder traversal of T can be determined from V.
(b) Inorder traversal of T can be determined from V.
(c) Postorder traversal of T can be determined from V.
(d) Root node of T can be determined from V.

[2024 (Set-2) : 2 M]

5.65 Which of the following statement(s) is/are TRUE for any binary search tree (BST) having n distinct integers?

(a) The maximum length of a path from the root node to any other node is $(n-1)$.
(b) An inorder traversal will always produce a sorted sequence of elements.
(c) Finding an element takes $O(\log_2 n)$ time in the worst case.
(d) Every BST is also a Min-Heap.

[2025 (Set-1) : 1 M]

5.66 The height of any rooted tree is defined as the maximum number of edges in the path from the root node to any leaf node.

Suppose a Min-Heap T stores 32 keys. The height of T is 5. (Answer in integer)

[2025 (Set-1) : 1 M]

5.67 Consider a binary tree T in which every node has either zero or two children. Let $n > 0$ be the number of nodes in T.

Which ONE of the following is the number of nodes in T that have exactly two children?

(a) $\frac{n-2}{2}$ (b) $\frac{n-1}{2}$
(c) $\frac{n}{2}$ (d) $\frac{n+1}{2}$

[2025 (Set-2) : 1 M]

5.68 Suppose the values 10, −4, 15, 30, 20, 5, 60, 19 are inserted in that order into an initially empty binary search tree. Let T be the resulting binary search tree. The number of edges in the path from the node containing 19 to the root node of T is 4. (Answer in integer)

[2025 (Set-2) : 1 M]

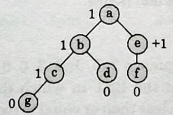
■■■■

Answers Trees

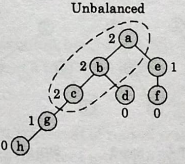
5.1	(b)	5.2	(c)	5.3	(b)	5.4	Sol.	5.5	(b)	5.6	(d)	5.7	(b)	5.8	(a)	5.9	(c)
5.10	(b)	5.11	(d)	5.12	(c)	5.13	(b)	5.14	(b)	5.15	(d)	5.16	(d)	5.17	(b)	5.18	(d)
5.19	(c)	5.20	(b)	5.21	(d)	5.22	(a)	5.23	(c)	5.24	(b)	5.25	(c)	5.26	(a)	5.27	(b)
5.28	(c)	5.29	(c)	5.30	(b)	5.31	(a)	5.32	(c)	5.33	(c)	5.34	(a)	5.35	(a)	5.36	(b)
5.37	(a)	5.38	(c)	5.39	(b)	5.40	(d)	5.41	(b)	5.42	(a)	5.43	(d)	5.44	(1)	5.45	(a)
5.46	(d)	5.47	(a)	5.48	(a)	5.49	(19)	5.50	(b)	5.51	(64)	5.52	(b)	5.53	(b)	5.54	(4)
5.55	(4.25)	5.56	(d)	5.57	(c)	5.58	(d)	5.59	(1)	5.60	(509)	5.61	(c)	5.62	(a)	5.63	(c)
5.64	(b)	5.65	(a, b)	5.66	(5)	5.67	(b)	5.68	(4)								

Explanations Trees

5.1 (b)

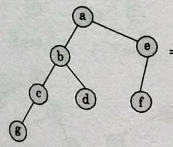


When a node is added as a child of g



Nodes a, b & c become unbalanced.
Therefore, option (b) is correct.

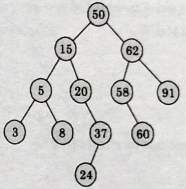
5.2 (c)



⇒ Postorder: gcdbfea

5.3 (b)

The binary search tree is



The number of nodes in the left subtree and right subtree of the root respectively is (7, 4)

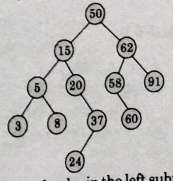
5.4 Sol.

Possible sequence:
(a) 61 52 14 17 40 43
(c) 10 65 31 48 37 43
(d) 81 61 52 14 41 43

Not possible:
(b) 2 3 50 40 60 43
(e) 17 77 27 66 18 43

5.5 (b)

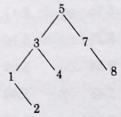
The binary search tree is as follows:



The number of nodes in the left subtree and right subtree of the root respectively is (7, 4). Therefore option (b) is correct.

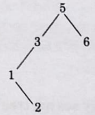
5.6 **(d)**

(a) 53124786



6 can not be child of 8.

(b) 53126487



4 can not be child of 6.

(c) 53241678

1 can not be child of 4 or 5.

5.7 **(b)**

A labelled rooted binary tree can not be constructed uniquely when inorder traversal is given along with post-order or pre-order traversal.

5.8 **(a)**

In n-ary tree the total number of vertices are

$$m = nx + 1$$

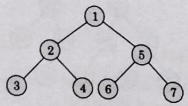
Where x: no. of internal nodes and let l be the number of leaf nodes.

$$x + l = nx + 1$$
$$l = (n-1)x + 1$$

So number of leaf nodes in n-ary tree with x internal nodes is $(n-1)x + 1$.

5.9 **(c)**

(1 (2 3 4) (5 6 7))



5.10 **(b)**

Because of complete binary tree option (b) is always correct.

5.11 **(d)**

In m-ary tree

Total number of nodes = m(internal nodes) + 1

Here, $n = 3i + 1$

Total nodes = internal nodes + leaf nodes

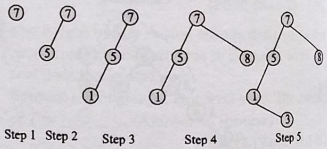
$$n = i + l$$
$$i = n - l$$
$$n = 3(n-l) + 1 \quad [\because n = 3i + 1 \ \& \ i = n - l]$$
$$n = 3n - 3l + 1$$
$$3l = 2n + 1$$
$$l = \frac{2n+1}{3}$$

Where l is leaf nodes.

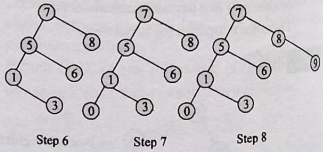
5.12 **(c)**

Input sequence 7, 5, 1, 8, 3, 6, 0, 9, 4, 2

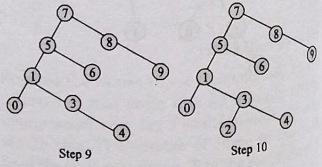
Construct the BST in which the value of left subtree is less then or equal to root and the value of right-subtree is greater than or equal to the root.



Step 1 Step 2 Step 3 Step 4 Step 5



Step 6 Step 7 Step 8



Step 9 Step 10

In order traversal 0 1 2 3 4 5 6 7 8 9

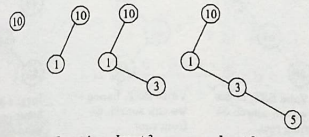
Note: The in-order traversal of a BST is in always sorted order.

5.13 **(b)**

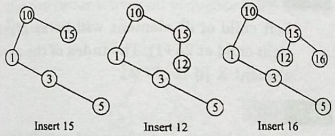
Insertion and deletion in AVL and Min heap will take O(logn). But in this problem we have to check before inserting that element is already there are not, to do this Min heap will take O(n). AVL will take O(logn).

5.14 **(b)**

The binary search tree is initially empty.



Insert 10 Insert 1 Insert 3 Insert 5

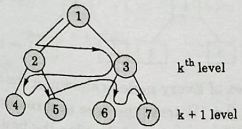


Insert 15 Insert 12 Insert 16

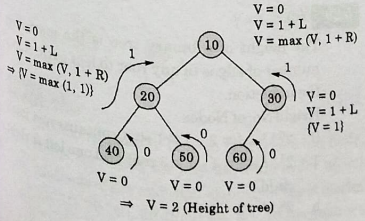
So the maximum height of BST = 3 [Distance from element 5 to root 10 is 3]

5.15 **(d)**

Level order traversal of rooted tree can be done by starting from root and visiting all node at k level before visiting any node at k + 1 level, which is nothing but breadth first search.



5.16 **(d)**

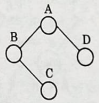


5.17 **(b)**

In a labeled binary tree preorder traversal produces prefix expression, postorder produces postfix expression and inorder produce infix expression. Preorder traversal root is first element and using inorder we can identity left subtree nodes and right subtree nodes for that root. So we can construct unique Binary tree using preorder and inorder similarly postorder and inorder. The postorder numbers assigned to the nodes have the useful property that the nodes in the subtree with root n are numbered consecutively from (postorder(n) – desc(n)) to postorder(n). To test if a vertex x is a descendent of vertex y then postorder(y) – desc(y) ≤ postorder(x) ≤ postorder(y)

A similar property holds for preorder and doesn't holds for inorder.

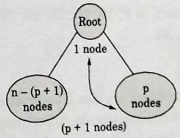
5.18 **(d)**



Inorder traversal is BCAD and preorder traversal is ABCD.

5.19 **(c)**

Number 1, 2, 3, ... n are inserted into BST in some order.



Since we know total nodes = n

So, Left (root) + 1 + p = n

Left (root) = n – p – 1

Left (root) = n – (p + 1)

So, root element will be = left element + 1

= n – (p + 1) + 1

= n – p – 1 + 1

= n – p

5.20 **(b)**

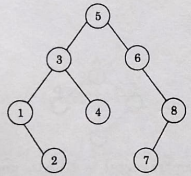
For h = 1 ⇒ Min nodes = 2

h = 2 ⇒ Min nodes = 3

So by putting in option we get option (b) is correct.

$2^{h-1}+1$.

5.21 **(d)**



5.22 **(a)**

The inorder traversal of a binary search tree is always is in sorted order or increasing order of a given sequence so the inorder traversal of the tree T is 9, 10, 15, 22, 23, 25, 27, 29, 40, 50, 60, 55

5.23 **(c)**

In a complete-k-ary tree. If every internal node has exacity k children then (k – 1) key contains no leaves. If there are n internal nodes, then number of leaves is n(k – 1) + 1.

5.24 **(b)**

Let number of vertices in binary tree = n

Number of edges = n – 1

So $n-1 = 1\times 5 + 2\times 10$

$n-1 = 5+20$

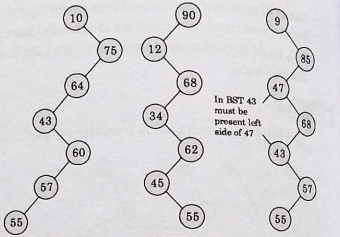
$\Rightarrow \quad n = 26$

Number of leaf nodes = 26 – 5 – 10

= 11

5.25 **(c)**

According to given options:



Valid BST, hence we can search 55 from root

Valid BST, hence we can search 55 from root

Invalid BST, hence cannot be valid sequence

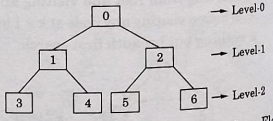
5.26 **(a)**

Left child of i^{th} element will be at 2*i+1 and right child at 2(i+1). The index of the parent of element X [i] is $\lfloor i/2 \rfloor$.

5.27 **(b)**

Since only root node does not satisfy the heap property, so applying one time heapify will make whole tree as heap. Heapify will take $O(\log_2 n)$ i.e. height of heap tree.

5.28 **(c)**



Index of Every node is shown in the box. Floor (log (i + 1)) draw the tree and realise that the last element at each level is the best choice to arrive at a conclusion.

5.29 **(c)**

The height of a binary tree is the maximum number of edges in any root to leaf path. Apply the induction.

Height	No. of Nodes
0	$2^{0+1}-1 = 2-1 = 1$ so it contains root itself
1	$2^{1+1}-1 = 4-1 = 3$ root and one left & right child
h	$2^{h+1}-1$

So maximum number of nodes in a binary tree of height h is $2^{h+1}-1$.

5.30 **(b)**

The maximum number of Binary search contain *n* noes

$$= \frac{1}{n+1}\binom{2n}{n}$$

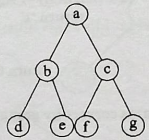
Given, $n = 3$

$$= \frac{1}{3+1}\,{}^{6}C_3 = \frac{1}{4}\cdot\frac{\lfloor 6}{\lfloor 3\cdot\lfloor 3}$$

$$= \frac{1}{4}\times\frac{6\times5\times4\times\lfloor 3}{3\times2\times1\times\lfloor 3} = 5$$

5.31 **(a)**

The in order traversal sequence is d b e a f c g and the pre order traversal sequence is a b d e c f g. So the tree is



In the post order traversal the sequence is d e b f g c a.

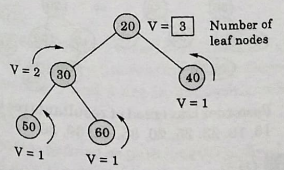
5.32 **(c)**

I[n – 1] + 1 = L

$$n = \frac{(L-1)}{I}+1 = \frac{41-1}{10}+1 = \frac{40}{10}+1$$

n = 4 + 1 = 5

5.33 **(c)**



5.34 **(a)**

In Binary Search Tree to traverse each node, while searching for any key only order of left elements of key and right element of key is matter i.e. 10, 20, 40, 50 [key] 90, 80, 70 because we need to search element 60, so order of elements less than 60 will be only 10, 20, 40, 50 and order of elements greater than 60 will be 90, 80, 70.

So number of ways are = Arrangement of 7 numbers such that 4 element order same and remaining 3 elements order also same

$$= \frac{7!}{3!4!} = 35$$

5.35 **(a)**

Height of AVL tree always O(log n), so searching of an element will take O(log n) but height of a BST can be O(n) since skew BST also possible. So searching an element will take O(n) time.

5.36 **(b)**

According to given data:

$n_1\times1 + n_2\times2 + n_3\times3 = 2e$ (using Handshaking lemma)

In binary tree number of nodes (n) = e + 1

$n = e+1$

$e = n-1$

i.e. $e = (n_1+n_2+n_3-1)$

$n_1\times1 + n_2\times2 + n_3\times3 = 2(n_1+n_2+n_3-1)$

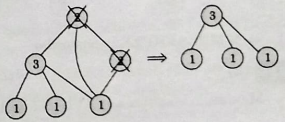
$n_1 + 2n_2 + 3n_3 = 2n_1 + 2n_2 + 2n_3 - 2$

$n_1 = n_3 + 2$

and $n_3 = n_1 - 2$

5.37 **(a)**

Since all node of degree 2 are removed after adding an edge between neighbour of nodes degree 2. Ex: Each node showing its degree.



It will not effect the number of nodes of degree 1 and degree 3.

So, according to Handshaking lemma.

$n_1\times1 + n_3\times3 = 2e$

We know, $n_3 = n_1 - 2$

$n_1\times1 + (n_1-2)\times3 = 2e$

$n_1 + 3n_1 - 6 = 2e$

$4n_1 - 6 = 2e$

$$e = \frac{4n_1-6}{2} = 2n_1 - 3$$

5.38 **(c)**

We have post order traversal and the tree is binary search tree so in order traversal of binary search tree is assending order.
Using these two we can construct binary search tree with O(nlogn) [we can apply binary search].

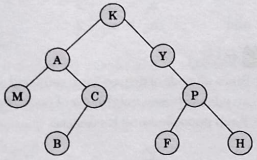
5.39 **(b)**

If we have a binary heap on elements and we wish to insert n more elements then we call a recursive call build heap which is responsible for constructing the heap.
The time spend by build heap is on the order of the sum over all vertices of the heights of the vertices. But at most $\lceil n/2^{i+1} \rceil$ vertices are of height i. So the total time spent by build heap is

$$\sum_{i=1}^{h} \frac{n_i}{2^i} = \theta(n).$$

5.40 **(d)**

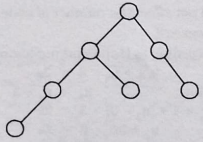
II. KAMCBYPFH (Preorder)
III. MABCKYFPH (Inorder)
If II is preorder and III is inorder then it is possible to construct binary tree.



⇒ I. MBCAFHPYK is postorder.

5.41 **(b)**

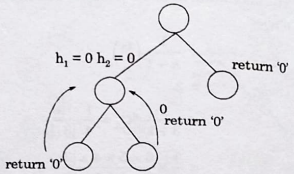
Maximum height of any AVL-tree with 7.



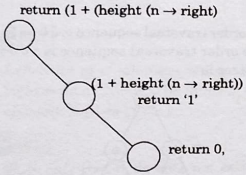
(There may be different way to draw AVL with 7 nodes).

5.42 **(a)**

e.g.
$h_1 = 1 + \max(0, 0)$, $h_2 = 0$,
return $1 + \max(h_1, h_2)$



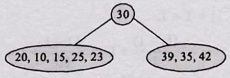
another e.g.

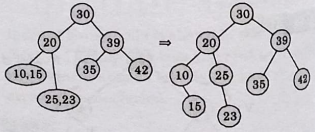


∴ Option (a) is correct by going through option.

5.43 **(d)**

Binary search tree and preorder is given
30, 20, 10, 15, 25, 23, 39, 35, 42



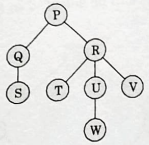


Postorder traversed of resultant tree is
15, 10, 23, 25, 20, 35, 42, 39, 30

5.44 **(1)**

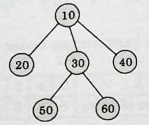
Upper bound on time required to determine the number of subtrees having exactly 4 nodes = $O(n)$
$= O(n^a \log^b n)$
⇒ $a = 1, b = 0$
∴ $a + 10b = 1$

5.45 **(a)**

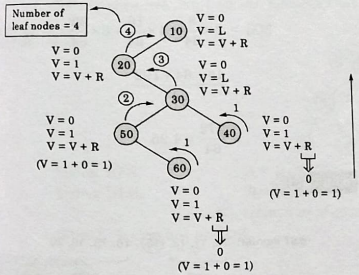


Inorder : (Left, root, mid, right)
∴ **Inorder is :** S Q P T R W U V

5.46 **(d)**



Left most child right sibling representation



5.47 **(a)**

The INORDER traversal of BST will always be sorted sequence (increasing order). Hence the sequences 1 and 4 are in increasing order.

5.48 **(a)**

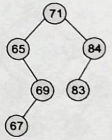
Height is defined as path length i.e. (height +1) will give number of levels.
In a binary tree of height h = 5:
Maximum number of nodes = $2^{h+1} - 1 = 2^{5+1} - 1$
$= 64 - 1 = 63$
Minimum number of nodes = 6 i.e. one node at every level.

5.49 **(19)**

n_0 : Number of leaf nodes (degree 0)
n_1 : Number of nodes with degree 1
n_2 : Number of nodes with degree 2
We have following two equations for binary tree
$n = n_0 + n_1 + n_2$ (where n is the total number of nodes in the tree) ...(1)
Accounting for total nodes as sum of those nodes which are children and the root which is not a child of any node,
we get
$n = n_1 \times 1 + n_2 \times 2 + 1 = n_1 + 2n_2 + 1$ ...(2)
Combining equations (1) and (2) we get
$n_0 + n_1 + n_2 = n_1 + 2n_2 + 1 \Rightarrow n_2 = n_0 - 1$
Therefore the number of nodes of degree two in any binary tree = number of leaf nodes −1.
$= 20 - 1 = 19$

5.50 **(b)**

71, 65, 84, 69, 67, 83 are inserted into empty BST. Binary search tree:



∴ 67 is present in the last level.

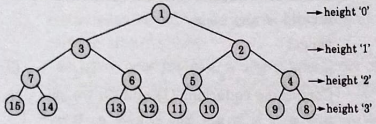
5.51 **(64)**

In question restriction on BST is height should be '6'. So we need 7 levels (given that root at height '0'). In creation of BST we have to use all element without repeatation.
At **1 level** = We have 2 choice i.e., either take 1 or 7.
At **2 level** = We have 2 choice for root 1 and 7 each. If 1 is root then 2 choice will be 6 and 2. If 7 is root then 2 choice will be 1 and 6.
At **3 level** = If we take 1 at root, 6 at 2^{nd} level than we have 2 choice i.e., 5 and 2 at 3^{rd} level. If we take 1 at root, 2 at 2^{nd} level than we have 2 choice i.e., 6 and 3 at 3^{rd} level. Similarly if we take 7 as root element.

So till 6th level, we have two choice at every level and for last level we left with only 1 element.
So number of BST with height 6
$= 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 1 = 2^6 = 64.$

5.52 (b)

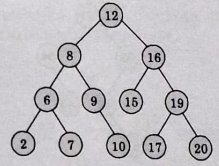
Since there are 15 nodes, hence the minimum height of the tree will be 3 (when tree will be balanced).



The maximum height will be when the tree is skew tree, which will give rise to height 14.

5.53 (b)

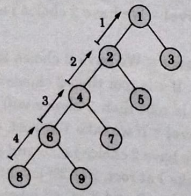
Preorder: 12, 8, 6, 2, 7, 9, 10, 16, 15, 19, 17, 20
Inorder: 2, 6, 7, 8, 9, 10, 12, 15, 16, 17, 19, 20
Tree will be,



Postorder will be, 2, 7, 6, 10, 9, 8, 15, 17, 20, 19, 16, 12

5.54 (4)

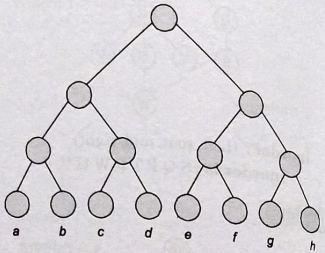
Postorder = 8, 9, 6, 7, 4, 5, 2, 3, **(1)** Root
Inorder = 8, 6, 9, 4, 7, 2, 5, 1, 3
Binary tree will be



So, height of binary tree will be 4.

5.55 (4.25)

Two nodes can be selected in $8 \times 8 = 64$ ways.



X : length between two nodes selected

X	0	1	2	3	4	5	6
Number of pairs	8	0	8	0	16	0	32

To find expected length between a and b is E[X]

$$E[X] = 2\times\frac{8}{64} + 4\times\frac{16}{64} + 6\times\frac{32}{64}$$

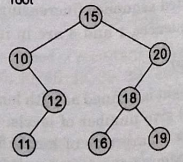
$$= \frac{16+64+192}{64}$$

$$= \frac{272}{64} = 4.25$$

5.56 (d)

BST inorder: 10, 11, 12, (15), 16, 18, 19, 20
LST RST

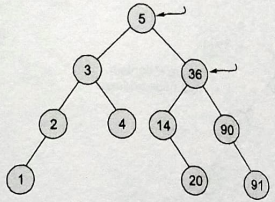
Preorder: (15), 10, 12, 11, 20, 18, 16, 19
↓ 1st root



Postorder: 11, 12, 10, 16, 19, 18, 20, 15

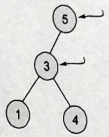
5.58 (d)

(i)



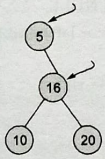
⇒ 5 is definitely lesser than max element

(ii)



⇒ 3 is definitely lesser than max element

(iii)



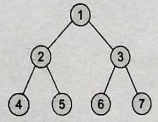
⇒ 5 is definitely lesser than max element

(iv)

No such element

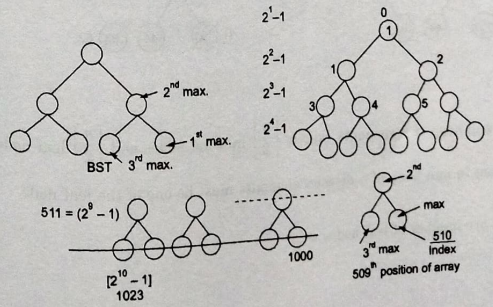
In any case, we can find one element that is less in just 2 seeks ⇒ θ(1).

5.59 (1)

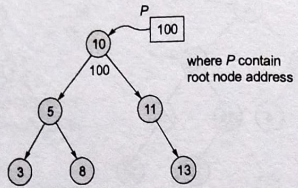


Using BFS, $A = \{1, 2, 3\}$
Using DFS, $B = \{1, 2, 4\}$
$|A - B|$ = Number of elements which are in A but not in B is only element {3}
So only 1 element present.

5.60 (509)



5.61 (c)



When foo is called with a pointer to the root node of the given binary tree then we reach the node until that node has left pointer and right pointer both pointing to NULL.

Whenever p → Left is NULL and p → Right is NULL then foo (p → left) and foo(p → right) both will return '0' and variable retval contain the following value.

$$\text{retval} = (p \rightarrow \text{val}) + \text{foo}(p \rightarrow \text{left}) + \text{foo}(p \rightarrow \text{right})$$

for node whose value is 3, retval = 3 + 0 + 0 = 3

for node whose value is 8, retval = 8 + 0 + 0 = 8

for node whose value is 5, retval = 5 + 3 + 8 = 16

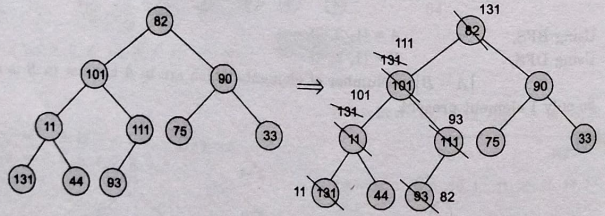
for node whose value is 13, retval = 13 + 0 + 0 = 13

for node whose value is 11, retval = 11 + 13 + 0 = 24

for node whose value is 10, retval = 10 + 16 + 24 = 50

5.62 (a)

Heapify: bottom-up construction
(Adjust internal nodes into max heap).
CBT of given 10 elements.



5.63 (c)

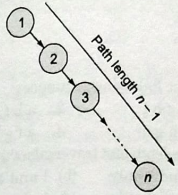
In binary min heap of n elements exactly $\left\lfloor \frac{n}{2} \right\rfloor$ internal nodes and $\left\lceil \frac{n}{2} \right\rceil$ leaf nodes.

Max element in min heap of n distinct elements must be one of the leaf node.

$\therefore$ $\left\lceil \frac{105}{2} \right\rceil$ are number of leaf nodes = 53

5.65 (a, b)

BST can be skewed tree also

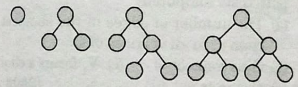


Inorder traversal of BST is sorted sequence. Worst case search cost of BST is $\theta(n)$. Not every BST is min-heap.

5.66 (5)

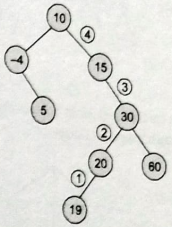
- Number of nodes in min heap with height h are 2^h to $2^{h+1} - 1$.
- 32 keys are minimum nodes in min heap of height 5.

5.67 (b)



Number of nodes (n)	Number of nodes of two childs
1	0
3	1
5	2
7	3
n	$\frac{n-1}{2}$

5.68 (4)



Number of edges from 19 to root node = 4.

■■■■