**Branch : CSE/IT** **Batch : Vijay (Hinglish)**

**Practice sheet _3**

# Data Structure

## Tree & Hashing

**[MCQ]**

**1.** 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 9 are-

(a) 1024, 9 (b) 1023, 10

(c) 511, 9 (d) 512, 10

**[MCQ]**

**2.** A strict k-ary tree T is a tree that contains exactly 0 or k children. The number of leaf nodes in tree T if there are exactly 'p' internal nodes is-

(a) (k – 1)p + 1 (b) pk + 1

(c) pk + 1 + p (d) None

**[NAT]**

**3.** Let T be a full binary tree with 4 leaves. (A full binary tree has every level full). Suppose two leaves x and y of T are chosen uniformly and independently at random. The expected value of the distance between x and y in T (i.e., the number of edges in the unique path between x and y) is (rounded off to 2 decimal places) _____.

**[MCQ]**

**4.** The number of leaf nodes in a rooted tree of n nodes, with each node having 0 or 2 children is-

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

(c) $\frac{n}{2}$ (d) n – 1

**[MCQ]**

**5.** Consider the following nested representation of binary trees: (X Y Z) indicates Y and Z are the left and right sub stress, 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 (2 3 4) 5 6) 7)

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

(d) (1 (2 3 NULL) (4 5))

**[MCQ]**

**6.** Consider the following two statements:

S1: It is possible to construct a binary tree uniquely whose post-order and pre-order traversals are given.

S2: It is possible to construct a binary tree uniquely whose in-order and pre-order traversals are given.

S3: It is possible to construct a binary tree uniquely whose post-order and level-order traversals are given.

Which of the following statement(s) IS/ARE INCORRECT?

(a) S1 only (b) S2 only

(c) S1 and S3 (d) S3 only

**[MCQ]**

**7.** 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

**[NAT]**

**8.** The pre-order traversal of a binary tree is 1, 2, 4, 7, 8, 3, 5, 6, 9. The in-order traversal of the same tree is 7 4 8 2 1 5 3 6 9. 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 ______.

**[MCQ]**

**9.** The post-order traversal of a binary tree is 9, 7, 4, 8, 2, 5, 1, 3, 6. The in-order traversal of the same tree is 9, 7, 8, 4, 5, 2, 6, 3, 1. The pre-order traversal of the above binary tree is-

(a) 1, 2, 4, 7, 9, 8, 5, 3, 6

(b) 1, 2, 4, 7, 8, 9, 5, 3, 6

(c) 1, 2, 3, 4, 5, 6, 7, 8, 9

(d) None of the above.

**[NAT]**

**10.** The number of labelled binary trees possible with the nodes-10, 30, 25, 40 is _______.

**[MCQ]**

**11.** Consider the following two statements:

**Statement P:** The last elements in the pre-order and inorder traversal of a binary search tree are always same.

**Statement Q:** The last elements in the pre-order and inorder traversal of a binary tree are always same.

Which of the following tree is/are CORRECT?

(a) Both P and Q (b) Neither P nor Q

(c) Q only (d) P only

**[MCQ]**

**12.** Consider the following function:

```
struct treenode{
struct treenode*left;
int data;
struct treenode *right;
};
int func (struct treenode *t){
if(t = NULL) return 1;
else if(t→left = NULL && t → right = NULL)
return 1;
else if
((t →left → data < t → data) && (t → right → data >
t-
>data))
return func(t→left) && func(t → right);
else
return 0;
}
```

Assume t contains the address of the root node of a tree. The function-

(a) Returns 1 if the given tree is a Binary Search Tree.

(b) Returns 0 if the given tree is a complete binary tree.

(c) Returns 0 if the given tree is a Binary Search Tree.

(d) Returns 1 if the given tree is a complete binary tree.

**[MCQ]**

**13.** Consider the following function: struct treenode {

```
struct treenode *left;
int data:
struct treenode *right;
};
struct treenode * f(struct treenode *t, int x)
{
    if(t=NULL) return NULL;
    elseif(x=t->data) return a;
    else if (x<t->data) return b;
    else return c;
    >
```

Assume t contains the address of the root node of a binary search tree. The function finds an element x in the BST and returns the address of the node if found. Which of the following statement(s) is/are CORRECT?

(a) a: NULL ; b: f(t->left, x); c: t->right, x)

(b) a: t; b: f(t->right, x); c: f(t->left, x)

(c) a: NULL ; b: f(t->right, x); c: f(t->left, x)

(d) a: t; b: f(t->left, x); c: f(t->right, x)

**[MCQ]**

**14.** Consider the following function:

```
struct treenode
{
    struct treenode *left;
    int data;
    struct treenode *right;
  };
int func(struct treenode *p)
{
    if(p==NULL) return 1;
    else if(p->right!=NULL) return 0;
    return func(p->left);
    }
```

Initially p contains the root node address of the tree, the function-

(a) Returns 1 if a binary tree is left-skewed.

(b) Returns 1 if a binary tree is right-skewed.

(c) Returns 1 if a binary tree is not right-skewed.

(d) None of the above.

**[MCQ]**

**15.** Consider the following function:

```
struct treenode
{
    struct treenode *left;
    int data;
    struct treenode *right;
  };
int func(struct treenode *t)
```

```
{
        if(t==NULL) return 0;
        elseif(t->left==NULL    &&    t->right==NULL)
        return 1;
        else
        return 1+func(t->left)+func(t->right);
}
```

Assume, t is a pointer to the root node of a binary tree, the function computes-

(a) Number of leaf nodes in the binary tree

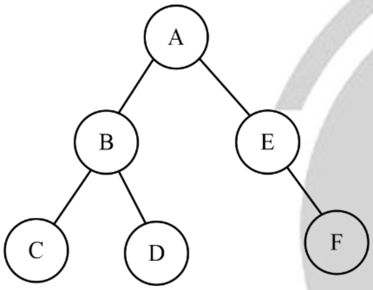
(b) Number of internal nodes in the binary tree

(c) Total number of nodes in the binary tree

(d) None of the above

**[MCQ]**

**16.** The given tree is passed to the following function:



```
void func(struct treenode *t)
{
   if(t)
   {
      printf("%d", t->data);
      func(t->right);
      printf("%d", t->data);
      func(t->left);
   }
}
```

The output string is-

(a) AEFFEBDDCCBA

(b) AEFFEABDDBCC

(c) AEFFEBDDCCBA

(d) None of the above

**[MCQ]**

**17.** Consider the following function:

```
struct treenode
{
        struct treenode *left;
        int data;
        struct treenode *right;
  };
void func(struct treenode *p){
while(p->left!=NULL) p=p->left;
printf("%d", p->data);
}
```

If the address of the root node of the BST is passed to p, the above function prints-

(Assume, the tree contains at least one node)

(a) The maximum element in the BST

(b) The ancestor of two leftmost leaf nodes

(c) The minimum element in BST

(d) None of the above

**[MCQ]**

**18.** Consider the following two statements:

P: The minimum number of nodes in a complete binary tree is $2^{h+1}$.

Q: A binary search tree is always a complete binary tree.

Which of the statement(s) is/are CORRECT?

(a) P only (b) Q only

(c) Both P and Q (d) Neither P nor Q

**[MSQ]**

**19.** Consider the following operations in a BST-

INSERT(23), INSERT(17), INSERT(25), INSERT(4), INSERT(21), INSERT(1), INSERT(7), DELETE(17), DELETE(23).

The post-order traversal of the resultant BST is-

(a) 1, 7, 4, 21, 25

(b) 1, 4, 7, 25, 21

(c) 1, 4, 21, 7, 25

(d) None of the above

**[NAT]**

**20.** Consider the following statements:

P: The accepted balanced factor in an AVL tree are –1,0 and +1.

Q: The height of an AVL tree with n nodes is given as $\lceil log2n \rceil$.

The number of INCORRECT statements is ______.

**[NAT]**

**21.** Consider the array given below:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 50 | 40 | 10 | 8 | 60 | 70 | 70 | 15 | 80 |

The minimum number of comparisons required to convert the above array into max heap is _______

**[MCQ]**

**22.** Consider a sequence of elements are inserted into a max-heap one after another as-

50, 40, 10, 5, 60, 70, 40, 15, 80

The resultant max-heap using bottom-up approach of build heap is-

(a) 80, 60, 70, 40, 50, 10, 40, 15, 5

(b) 80, 70, 60, 50, 40, 10, 40, 5, 15

(c) 80, 70, 60, 50, 40, 40, 15, 10, 5

(d) None of the above

**[MCQ]**

**23.** Consider the following two statements:

P: The number of comparisons required to find the minimum element in a min heap of n elements is n-1.

Q: Only one comparison is required to find the minimum element in a max heap of n elements.

Which of the following is/are CORRECT?

(a) P only

(b) Q only

(c) Both P and Q

(d) Neither P nor Q

**[MCQ]**

**24.** 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 pre-order 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

**[MCQ]**

**25.** A Binary Search Tree (BST) stores values in the range 37 to 573. Consider the following sequence of keys.

I.81, 537, 102, 439, 285, 376, 305

II.52, 97, 121, 195, 242, 381, 472

III.142, 248, 520, 386, 345, 270, 307

IV.550, 149, 507, 395, 463, 402, 270

Suppose the BST has been unsuccessfully searched for key 273. Which all of the above sequences list nodes in the order in which we could have encountered the min the search?

(a) I and III (b) II and III

(c) III and IV (d) III only

**[MCQ]**

**26.** A Priority-Queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is given below: 10, 8, 5, 3, 2. Two new elements ‘1’ and ‘7’ are inserted in the heap in that order. The level-order traversal of the heap after the insertion of the elements is:

(a) 10, 8, 7, 3, 2, 1, 5

(b) 10, 8, 7, 2, 3, 1, 5

(c) 10, 8, 7, 3, 2, 5, 1

(d) None of the above

**[NAT]**

**27.** Consider a hash function that distributes keys uniformly. The hash table size is 2024. After hashing of how many keys will the probability that any new key hashed collides with an existing one exceed 0.75?

____________

**[MCQ]**

**28.** Suppose we are given n keys, m hash table slots, and two simple uniform hash functions $h_1$ and $h_2$. Further suppose our hashing schemes uses $h_1$ for the even keys and $h_2$ for the odd keys. What is the expected number of keys in a slot?

(a) $\frac{n}{m}$ (b) $\frac{m}{n}$

(c) $\frac{2n}{m}$ (d) $\frac{m}{2n}$

**[MCQ]**

**29.** Consider a hash table with 11 slots. The hash function is h(k) = k mod 11. The collisions are resolved by chaining. The following 11 keys are inserted in the order: 28, 19, 15, 20, 33, 30, 42, 63, 60, 32, 43. The maximum, minimum, and average chain lengths in the hash table, respectively, are-

(a) 3, 0, 1 (b) 3, 3, 3

(c) 3, 0, 2 (d) 4, 0, 1

**[MCQ]**

**30.** A hash table of length 8 uses open addressing with hashfunction h(k)= 2 + k mod 8, and linear probing. After inserting 5 values into an empty hash table, the table isas shown below.

| | |
|---|---|
| 0 | |
| 1 | |
| 2 | 64 |
| 3 | 41 |
| 4 | 57 |
| 5 | 72 |
| 6 | |
| 7 | 29 |

How many different insertion sequences of the key values using the same hash function and linear probingwill result in the hash table shown above?

(a) 10 (b) 9

(c) 15 (d) 8

# Answer Key

| | |
|---|---|
| **1.** | **(b)** |
| **2.** | **(a)** |
| **3.** | **(2.5)** |
| **4.** | **(a)** |
| **5.** | **(c)** |
| **6.** | **(c)** |
| **7.** | **(d)** |
| **8.** | **(3)** |
| **9.** | **(a)** |
| **10.** | **(336)** |
| **11.** | **(b)** |
| **12.** | **(a)** |
| **13.** | **(d)** |
| **14.** | **(a)** |
| **15.** | **(c)** |
| **16.** | **(b)** |
| **17.** | **(c)** |
| **18.** | **(d)** |
| **19.** | **(a, b, c)** |
| **20.** | **(0)** |
| **21.** | **(10)** |
| **22.** | **(b)** |
| **23.** | **(d)** |
| **24.** | **(a)** |
| **25.** | **(d)** |
| **26.** | **(a)** |
| **27.** | **(1518)** |
| **28.** | **(a)** |
| **29.** | **(a)** |
| **30.** | **(c)** |