COURSe : DAC - FEB 2020

Subject : Algorithms and DS   Date : 4.1.2021
Time: 10 to 11   Duration: 1 hour

Q. No. 1

The Time Complexity of Enqueue is
   a.   O(1)
   b.   O(0)
   c.   O(n)
   d.   O(n-1)

Q. No. 2

The Time Complexity of Pop Operation in stack is
   a.   O(1)
   b.   O(0)
   c.   O(n)
   d.   O(n-1)

Q. No. 3

A Stack is used in
   a.   Local variable tracking
   b.   Syntax analyzer
   c.   Both a) and b)
   d.   None of the above

Q. No. 4

Asymptotic Time complexity to add an element in the linked list
   a.   O(1)
   b.   O(0)
   c.   O(n)
   d.   O(n-1)

Q. No. 5

In a Double Ended Queue (Dequeue) if these operations are performed on an empty queue what would the Queue comprise at the end?

```
int main(){
        Insert_Front(1);
        Insert_Front(2);
        Insert_Rear(3);
        Delete_Front();
        Insert_Rear(4);
        Insert_Rear(1);
        Delete_Rear();
        Insert_Rear(5);
        Display_list();
}
```
   a.   1 3 1 5
   b.   2 3 4 5
   c.   2 3 4 1
   d.   1 3 4 5

Q. No. 6

With minimum time complexity to sort an unsorted linked list we use
   a.   Heap sort
   b.   Linear Sort
   c.   Bubble sort
   d.   Merge Sort

Q. No. 7

The Time Complexity to count the elements in a linked list is
   a.   O(1)
   b.   O(0)
   c.   O(n)
   d.   O(n-1)

Q. No. 8

The output of this prefix notation is
   - + 10 / 20 10 10
   a.   2
   b.   5
   c.   10
   d.   7

Q. No. 9

Converting prefix to infix time complexity is
   a.   O(1)
   b.   O(0)
   c.   O(n)
   d.   O(n-1)

Q. No. 10

When you convert infix to postfix, When an operator is read then it is placed in
   a.   Output
   b.   Stack
   c.   Temp variable
   d.   Any of the above

Q. No. 11

The Postfix form a+b*c+d is
   a.   abc*+d+
   b.   abc+*d+
   c.   abc++d*
   d.   ab+cd+*

Q. No. 12

The Infix form of abc-*d-
   a.   a*(b-c)-d
   b.   (a-b)*c-d
   c.   a-(b-c)*d
   d.   None of the above

Q. No. 13

Which one the following option gives the best definition of a collision in a hash table?
    a. Two entries are identical except for their keys
    b. Two entries with different data have the exact same key
    c. Two entries with different keys have the same exact hash value
    d. Two entries with the exact same key have different hash values

Q. No. 14

In a Binary search tree if pre-order traversal produces 10 5 16 then post order is

    a. 5 16 10
    b. 10 16 5
    c. 5 10 16
    d. 16 5 10

Q. No. 15

In a Binary Search tree if post-order traversal produces 6 2 1 then pre-order is
    a. 1 2 6
    b. 6 1 2
    c. 1 6 2
    d. 2 1 6

Q. No. 16

A Binary Search tree contains three nodes then the number leaf or leaves present is/are
    a. 1
    b. 2
    c. 1 or 2
    d. 3

Q. No. 17

Inorder traversal of a binary search tree
    a. Traverses is Increasing Order
    b. Traverses is Non-Increasing Order
    c. Traveses Randomly
    d. None of the above

Q. No. 18

A Balanced full Binary Tree with 8 leaves has _____ nodes

    a. 16
    b. 15
    c. 17
    d. 12

Q. No. 19

A Binary search tree with 8 null nodes has _____ nodes
    a. 7
    b. 16
    c. 9
    d. 10

Q. No. 20

What Data structure is used in RDBMS to store data?
- a. Binary Search Tree
- b. Balanced Binary Tree
- c. B+ tree
- d. B Tree

Q. No. 21

Which algorithm is used in solving the Eight Queens problem?
- a. Recursion
- b. Backtracking
- c. krushkal's algorithm
- d. None of the above

Q. No. 22

The Multiplication method of hashing function uses
- a. h(k) = floor(m(kA mod 1))
- b. h(k) = ceil(m(kA mod 1))
- c. h(k) = floor(kA mod m)
- d. h(k) = ceil(kA mod m)

Q. No. 23

Convert the following infix expression into their Prefix form
(A^B)/(C*D)
- a. / ^ A B * C D
- b. A B ^ C D * /
- c. A ^ B C D * /
- d. None of the above

Q. No. 24

Convert the following infix expression into their Prefix form.
A+B^C*D\E+F*(G-H)

- a. ++A*^BCDE*F-GH
- b. AB^CDE*\+FGH*-+
- c. -*+ABC^D*E\+FGH-*+
- d. None of the above

Q. No. 25

Using stack algorithm to convert the expression 4+3*(6*3-12) to postfix notation, The maximum number of symbols (operators & parenthesis) that will appear on the stack AT ONE TIME during the conversion of this expression is
- a. 1
- b. 2
- c. 3
- d. 4

Q. No. 26

If the characters 'D', 'C', 'B', 'A' are placed in a queue (in that order), and then removed one at a time, in what order will they be removed?

a. ABCD
b. ABDC
c. DCAB
d. DCBA

Q. No. 27

Consider the following c++ function

```
void test_a(int n) {
        cout << n << " ";
        if (n>0)
        test_a(n-2);
}
```
What is printed by the call test_a(4)?
a. 0 2 4
b. 2 4
c. 4 2
d. 4 2 0

Q. No. 28

Consider the following function: What is printed by the call test_b(4)?

```
void test_b(int n) {
        if (n>0)
         test_b(n-2);
        cout << n << " ";
}
```

a. 0 2 4
b. 2 4
c. 4 2
d. 4 2 0

Q. No. 29

You What is this code doing in a binary search tree?

```
void min(BST node)
{
        while(node.left() != null) {
                node = root.left();
        }
        cout<< node.data();
}
```

a. Finds the maximum element
b. Find the minimum element
c. Searching for a particular element
d. In-order traversal

Q. No. 30

If you want to implement the heterogeneous linked list, what pointer type will you use?

    a. Normal pointer
    b. null pointer
    c. void pointer
    d. None of the above

Q. No. 31

To retrieve a value stored in a hash table.

    a. Do a linear search on the table
    b. Do a binary search on the table.
    c. Hash the key and then locate the associated record.
    d. Construct a binary search tree from the table and search the tree.

Q. No. 32

Point mutations of strings s1 into s2 are
    a. change a letter
    b. insert a letter or
    c. delete a letter
    d. Any one of the above

Q. No. 33

I have implemented the queue with a linked list, keeping track of a front pointer and a rear pointer. Which of these

pointers will change during an insertion into an EMPTY queue?

    a. Neither changes
    b. Only front_ptr changes
    c. Only rear_ptr changes
    d. Both change

Q. No. 34

What this code is doing in a Binary search tree?

```
void do_job(BST node){
        if(node!=NULL) {
                do_job (node.left());
                do_job (node.right());
                cout<<Node.data;
        }
}
```

    a. Traversing post-order
    b. Traversing pre-order
    c. Traversing in-order
    d. Finding the dept

Q. No. 35

What is the recursive traversing of post-order traversal

    a.   traverse the left sub-tree, visit the root node and traverse the right sub-tree
    b.   visit the root node, traverse the left sub-tree, and traverse the right sub-tree
    c.   traverse the left sub-tree, traverse the right sub-tree, and visit the root node
    d.   None of the above

Q. No. 36

In-order, pre-order and post-order can be applied to
    a.   any trees
    b.   only binary trees
    c.   any trees other than binary trees
    d.   None of the above

Q. No. 37

Which of the following are not data structures?
    a.   Stack
    b.   Queue
    c.   linked list
    d.   groups

Q. No. 38

A queue structure would require
    a.   head pointer to remove an existing node
    b.   tail pointer to add to a new node
    c.   both (a) and (b)
    d.   None of the above

Q. No. 39

Which of the following are not related to stack?
    a.   Push
    b.   Pop
    c.   LIFO
    d.   FIFO

Q. No. 40

Pick the odd man out of searching
    a.   linear search
    b.   binary search
    c.   backward search
    d.   none of the above