Data Structure Tutorial 1

1. Which of the following is the applications of stack?
a) A parentheses balancing program b) Tracking of local variables at run time
c) Compiler Syntax Analyzer d) All of the mentioned

2. Which data structure is required for Breadth First Traversal on a graph?
a) Stack b) Array c) Queue d) Tree

3. In which data structure, elements can be inserted or deleted at/from both the ends but not in the middle?
a) Queue b) Circular queue c) Dequeue d) Priority queue

4. What data structure would you mostly likely see in a non recursive implementation of a recursive algorithm?
a) Linked List b) Stack c) Queue d) Tree

5. Which of the following is not an inherent application of stack?
a) Reversing a string b) Evaluation of postfix expression
c) Implementation of recursion d) Job scheduling

6. In linked list implementation of a queue, which of these pointers will change during an insertion into a NONEMPTY queue?
a) Only front pointer b) Only rear pointer c) Both front and rear pointer d) None

7. What is the condition for a queue to be empty in linked list implementation?
a) FRONT is null b) REAR is null c) LINK is empty d) None of the mentioned

8. In which type of expression, operator succeeds its operands?
a) Infix Expression b) Prefix Expression c) Postfix Expression d) None of the mentioned

9. Which of the following statements are correct with respect to Singly Linked List(SLL) and Doubly Linked List(DLL)?
a) Complexity of Insertion and Deletion at known position is O(n) in SLL and O(1) in DLL
b) SLL uses lesser memory per node than DLL
c) DLL has more searching power than SLL
d) All of the mentioned

10. What is the value of the postfix expression 6 3 2 4 + – *:
a) Something between -5 and -15 b) Something between 5 and -5
c) Something between 5 and 15 d) Something between 15 and 100

11. What is the time complexity of enqueue operation?
a) O(logn) b) O(nlogn) c) O(n) d) O(1)

12. Suppose a stack is to be implemented with a linked list instead of an array. What would be the effect on the time complexity of the push and pop operations of the stack implemented using linked list (Assuming stack is implemented efficiently)?
a) O(1) for insertion and O(n) for deletion b) O(1) for insertion and O(1) for deletion
c) O(n) for insertion and O(1) for deletion d) O(n) for insertion and O(n) for deletion
13. The process of accessing data stored in a serial access memory is similar to manipulating data on a _____?

a) Heap b) Binary Tree c) Array d) Stack

14. What is the reason for using a "circular queue" instead of a regular one?
a) the running time of enqueue() is improved. b) reuse empty spaces.
c) you can traverse all the elements more efficiently. d) None of the above.

15. The stack data type is restrictive in a sense that you cannot:
a) remove the top element b) insert at the top
c) look at the top element d) remove the bottom element

16. A priority queue can efficiently implemented using which of the following data structures? Assume that the number of insert and peek (operation to see the current highest priority item) and extraction (remove the highest priority item) operations are almost same.
a) Array b) Linked List c) Heap Data Structures d) None of the above

17. In linked list implementation of queue, if only front pointer is maintained, which of the following operation take worst case linear time?
a) Insertion b) Deletion c) To empty a queue d) Both a) and c)

18. Which one of the following is an application of Queue Data Structure?
a) When a resource is shared among multiple consumers.
b) When data is transferred asynchronously (data not necessarily received at same rate as sent) between two processes
c) Load Balancing
d) All of the above

19. Here is an infix expression: 8 + 2*(6*4-15). Suppose that we are using the usual stack algorithm to convert the expression from infix to postfix notation. What is the maximum number of symbols that will appear on the stack at one time during the conversion of this expression?
a) 1 b) 2 c) 3 d) 4

20. The postfix form of the expression (A+ B)*(C*D- E)*F / G is?
a) AB+ CD*E – FG /** b) AB + CD* E – F **G /
c) AB + CD* E – *F *G / d) AB + CDE * – * F *G /

21. The prefix form of A-B/ (C * D ^ E) is?
a) -/*^ACBDE b) -ABCD*^DE c) -A/B*C^DE d) -A/BC*^DE

22. Consider the usual algorithm for determining whether a sequence of parentheses is balanced. The maximum number of parentheses that appear on the stack at any time when the algorithm analyzes: (()(())(())) are:
a) 1 b) 2 c) 3 d) 4 or more

23. Consider a small circular linked list. How to detect the presence of cycles in this list effectively?
a) Keep one node as head and traverse another temp node till the end to check if its ‘next points to head.
b) Have fast and slow pointers with the fast pointer advancing two nodes at a time and slow pointer advancing by one node at a time.
c) Cannot determine, you have to pre-define if the list contains cycles.
d) None of the mentioned

24. Minimum number of queues to implement stack is _____.

a) 3 b) 4 c) 1 d) 2

25. A normal queue, if implemented using an array of size MAX_SIZE, gets full when:
a) Rear = MAX_SIZE – 1 b) Front = (rear + 1)mod MAX_SIZE
c) Front = rear + 1 d) Rear = front

26. In a circular queue, how do you increment the rear end of the queue?
a) rear++ b) (rear+1) % CAPACITY c) (rear % CAPACITY)+1 d) rear--

27. Which of the following is true about linked list implementation of queue?
a) In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end.
b) In push operation, if new nodes are inserted at the beginning, then in pop operation, nodes must be removed from the beginning
c) In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from end.
d) None of the mentioned

28. What is the time complexity of searching for an element in a circular linked list?
a) O(n) b) O(nlogn) c) O(1) d) None of the mentioned

29. Which of the following is false about a circular linked list?
a) Every node has a successor
b) Time complexity of inserting a new node at the head of the list is O(1)
c) Time complexity for deleting the last node is O(n)
d) None of the mentioned

30. What is the time complexity of deleting from the rear end of the dequeue implemented with a singly linked list?
a) O(nlogn) b) O(logn) c) O(n) d) $O(n^2)$

31. Assume that the operators +, -, × are left associative and ^ is right associative. The order of precedence (from highest to lowest) is ^, x , +, -. The postfix expression corresponding to the infix expression a + b× c - d ^ e ^ f is
a) abc× + def ^ ^ - b) abc× + de ^ f ^ - c) ab + c× d - e ^ f ^ d) - + a× bc ^ ^ def

32. Given an empty queue Q, what does it look like after the following operations?

```
Q.enqueue(5)
Q.enqueue(2)
Q.dequeue()
Q.enqueue(3)
Q.dequeue()
```

a) 3 b) 5 c) 9 d) none of the abov.

33. Consider the following operation performed on a stack of size 5.

```
Push(1);
Pop();
Push(2);
Push(3);
Pop();
Push(4);
Pop();
Pop();
Push(5);
```

After the completion of all operation, the number of elements present in stack are:

a) 1 b) 2 c) 3 d) 4

34. Given a 5 element stack S (from top to bottom: 2, 4, 6, 8, 10), and an empty queue Q, remove the elements one-by-one from S and insert them into Q, then remove them one-by-one from Q and re-insert them into S. S now looks like (from top to bottom).
a) 2, 4, 6, 8, 10 b) 10, 2, 4, 6, 8 c) 10, 8, 6, 4, 2 d) none of the above.

35. After performing these set of operations, what does the final list look contain?

```
InsertFront(10);
InsertFront(20);
InsertRear(30);
DeleteFront();
InsertRear(40);
InsertRear(10);
DeleteRear();
InsertRear(15);
display();
```

a) 10 30 10 15 b) 20 30 40 15 c) 20 30 40 10 d) 10 30 40 15

Solution: 1) d. 2) c. 3) c. 4) b. 5) d. 6) b. 7) a. 8) c. 9) d. 10) d. 11) d. 12) b. 13) d. 14) b. 15) d. 16) c. 17) a. 18) d. 19) d. 20) c. 21) c. 22) c. 23) b. 24) d. 25) a. 26) b. 27) a. 28) a. 29) b. 30) c. 31) a. 32) a. 33) a. 34) c. 35) d.