BIRLA INSTITUTE OF TECHNOLOGY
MESRA – 835215, RANCHI, INDIA

UG

Name: …… Roll No.: ……..

Branch: …… Signature of Invigilator: ……

Semester: IVth / VIth Date: 30/04/2022 (MORNING)

Subject with Code: **CS275 FUNDAMENTAL OF DATA STRUCTURE**

Marks Obtained	Section A (30)	Section B (20)	Total Marks (50)

INSTRUCTION TO CANDIDATE

1. The booklet (question paper cum answer sheet) consists of two sections. First section consists of MCQs of 30 marks. Candidates may mark the correct answer in the space provided / may also write answers in the answer sheet provided. The Second section of question paper consists of subjective questions of 20 marks. The candidates may write the answers for these questions in the answer sheets provided with the question booklet.
2. The booklet will be distributed to the candidates before 05 minutes of the examination. **Candidates should write their roll no. in each page of the booklet.**
3. Place the Student ID card, Registration Slip and No Dues Clearance (if applicable) on your desk. All the entries on the cover page must be filled at the specified space.
4. Carrying or using of mobile phone / any electronic gadgets (except regular scientific calculator)/chits are strictly prohibited inside the examination hall as it comes under the category of unfair means.
5. No candidate should be allowed to enter the examination hall later than 10 minutes after the commencement of examination. Candidates are not allowed to go out of the examination hall/room during the first 30 minutes and last 10 minutes of the examination.
6. Write on both side of the leaf and use pens with same ink.
7. The medium of examination is English. Answer book written in language other than English is liable to be rejected.
8. All attached sheets such as graph papers, drawing sheets etc. should be properly folded to the size of the answer book and tagged with the answer book by the candidate at least 05 minutes before the end of examination.
9. The door of examination hall will be closed 10 minutes before the end of examination. Do not leave the examination hall until the invigilators instruct you to do so.
10. Always maintain the highest level of integrity. Remember you are a BITian.
11. Candidates need to **submit the question paper cum answer sheets before leaving the examination hall**.

SET 1

Subject:CS275 FUNDAMENTAL OF DATA STRUCTURE (OE)

Sem: IV/VI OE Branch: BTECH OE EEE/ECE/MECH/PROD/CIVIL/BIOTECH/CHEM/ P&P

Each MCQ is 1 mark each.

1. ---------- may be defined as a finite sequence of instructions each of which has a clear meaning and can be performed with a finite amount of effort in a finite length of time.

a> Algorithm
b> Data structure
c> Apriori analysis
d> ADT

2. _____ means operations of the algorithm must be basic enough to be put down on pencil and paper.

a>Effectiveness

b> Finiteness

c>Generality

d>Definiteness

3. A _____ refers to the type of values that variables in a programming language hold.

a>data type

b> data object

c>type

d>stack

4. _____ and _____ are non linear data structures.

a> Trees and graphs
b> Trees and linked lists
c> Graphs and stacks
d> Stacks and queues

5. Data structures can be classified as _____ and _____

a> Linear and non linear
b> Stacks and queues
c> Linked and non linked
d> Structured and ADT

6. The number of times a statement is executed in the program is _____
   a> Apriori estimation
   b> Big Oh
   c> Omega
   d> Theta
7. For f(n)= $2n^2+n$,
   a> least upper bound is C=3, g(n)= n^2 for n>=1
   b> least upper bound is C=2, g(n)= n^3 for n>=1
   c> least upper bound is C=3, g(n)= n^3 for n<1
   d> none of above
8. Omega means_____
   a>best case
   b> worst case
   c>averagecase
   d>mean case
9. Consider the following algorithm:

Function factorial(n)

If (n=1) then factorial=1;

Else factorial=n*factorial(n-1);

End factorial

The time complexity of the above algorithm is

a>O(n)

b>O(n^2)

c> O(n^3)

d>O(n^4)

10. The compiler fills the array_____ by _____.

a>row by row

b>column by row

c>row by column

d>declaration by declaration

11. The postfix expression of a+b*c-d is
    a> abc*+d-
    b>a+bc*d-
    c>abc+*d-

12. Recursion makes use of
    a>stack
    b>queue
    c>tree
    d>graph
13. The following is checked in what situation?

If (rear==n) then queue_full;

a>insert operation on a queue

b>delete operation on a queue

c>insert operation on a stack

d>delete operation on a stack

14. The following is checked in what situation?

If (top==n) then stack_full;

a>push

b>pop

c>enqueue

d>dequeue

15. The following is checked in what situation?
    If (top==0) then stack_empty
    a>pop
    b>push
    c>dequeue
    d>enqueue
16. The following details are available about an array called RESULT. Find the address of RESULT[17].
    Base address: 520
    Index range: 1:20
    Array type:Real
    Size of memory location: 4 bytes
    a> 584
    b> 564
    c> 536
    d> 588
17. For the following array B, compute
    (i) The dimension of B
    (ii) The space occupied by B in the memory
    (iii) The address of B[7,2]

Array: B Column index: 0:5 Base address: 1003

Size of memory location: 4 bytes Row index: 0:15

a>96,384,1047
b>95, 384,1045
c>96,384,1045
d>16,384,1045

18. Two ways to implement a stack are
    a>linked list and array
    b>array and ADT
    c>linked list and queues
19. Following is a pseudocode of a series of operations on a stack S. PUSH(S,X) pushes an element X into S, Pop (S,X) pops out an element from stack S as X, Print(X) displays the variable X and Emptystack(S) is a Boolean function which returns true if S is empty and False otherwise. What is the output of following code?
    1. X:=30;
    2. Y:=15;
    3. Z:=20;
    4. Push(S,X);
    5. Push (S,40);
    6. Pop (S, Z);
    7. Push(S,Y);
    8. Push (S,30);
    9. Push (S,Z);
    10. Pop(S,X);
    11. Push(S,20);
    12. Push(S,X);
    13. While not Emptystack(S) do
    14. Pop(S,X);
    15. Print (X);
    16. End

    a> 40 20 30 15 30
    b> 40 30 20 15 30
    c> 30 20 40 15 30
    d> 30 30 40 15 30
20. DEQ[0:4] is an output restricted deque implemented as a circular array and LEFT and RIGHT indicate the ends of the deque as shown below. Insert('xx',[LEFT|RIGHT]) indicates the insertion of the data item at the left or right end as the case may be, and DELETE() deletes the item from left end only.

    DEQ:

[1]	[2]	[3]	[4]	[5]	[6]
	C1	A4	Y7	N6	

    LEFT:2

    RIGHT: 5

    Execute the following insertions and deletions on DEQ:

i)Insert ('S5', LEFT)

ii)Insert ('K9',RIGHT)

iii)DELETE()

iv)Insert('V7', LEFT)

v)Insert('T5', LEFT)

After all the insertions and deletions, what would the deque look like?

a>

[1]	[2]	[3]	[4]	[5]	[6]
V7	C1	A4	Y7	N6	K9

LEFT:1

RIGHT:6

b>

[1]	[2]	[3]	[4]	[5]	[6]
T5	C1	A4	Y7	N6	K9

LEFT:1

RIGHT:6

c>

[1]	[2]	[3]	[4]	[5]	[6]
S5	C1	A4	Y7	N6	K9

LEFT:1

RIGHT:6

21. DELETE_DL(P,X)

```
If(X==P) then ABANDON_DELETE;
Else
(RLINK(LLINK(X))=RLINK(X);
LLINK(RLINK(X)=LLINK(X);
Call Return (X);
End DELETE_DL;
```

The above procedure

a> Deletes node X from a headed circular doubly linked list P

b> Deletes node P from a headed circular doubly linked list X

c> Deletes nodes P and X from a circular linked list

22. Choose the correct procedure for deletion from a singly linked list with previous node as Nodex.

a>Procedure deletesl(start, Nodex)

```
If start=nil then call abandon_delete;
else
{
temp=Nodex->link;
Nodex->link=temp->link;
Call return temp;
End deletesl
```

b> Procedure deletesl(start, Nodex)

```
If start=nil then call abandon_delete;
else
{
Nodex->link=temp->link;
temp=Nodex->link
```

Call return temp;
End deletesl
c> >Procedure deletesl(start, Nodex)
If start=nil then call abandon_delete;
else
{
Nodex->link=temp->link;
Call return temp;
End deletesl

23. When last node points to first node, it is called
    a>circular linked list
    b>singly linked list
    c>doubly linked list
24. To delete a node , we don't need to specify a predecessor in
    a>circular linked list
    b>singly linked list
    c>all linked lists
25. Doubly linked list has a
    a>right link and a left link
    b>right link
    c>left link
26. Doubly linked list is expensive storage wise
    a> True
    b> False
27. _____ permits forward and backward movement
    a>Doubly linked list
    b>singly linked list
    c>all linked lists
28. Is the following Pop operation on a linked stack correct?
    If top==0 then call linkstack_empty;
    else
    {
    Temp=top
    Item=top->data
    top=top->link
    }
    Call return Temp

    a>Yes
    b>No
29. Is the following procedure correct for insertion into a linked queue?
    Call getnode(X);
    X->data=item;
    X->link=nil;
    If (front==0) then front=Rear=X;

```
else
{
Rear->link=X;
Rear=X;
}
```

a> Yes
b> No

30. What operations can we do with a linked list?
    a>insert, remove, traverse, search, concatenate
    b>insert and delete only
    c>traverse and concatenate only

Q1. State the properties of Binary Search Tree. Construct the Binary Search tree by inserting the elements in the following order.

23, 25, 28, 27, 15, 9, 19, 65, 97, 71, 5, 12, 16, 13

What will be final structure of the constructed tree after deleting 12 and 65? [5]

Q2: Explain various types of Tree traversal with an example. [5]

Q3. Explain the Depth First Search and Breadth First Search algorithm with an example. State the differences between DFS and BFS. [5]

Q4. Write the algorithm for heapsort. Sort the following elements using heapsort algorithm.

7, 5, 2,1,3,6,4,18,9,11,15,25,22 [5]

Explain each steps clearly.