Unit-1 Introduction

Short Questions:-

1. Define data structure.
2. Define algorithm.
3. List out areas in which data structures are applied.
4. List out structure of algorithm.
5. List out properties of algorithm.
6. List out the steps involved in the development of an algorithm.
7. Define Abstract Data Type.
8. List out the classification of data structure.
9. What is linear/primitive data structure?
10. What is non-linear/non-primitive data structure?
11. Gives the names of linear data structure.
12. Gives the names of non-linear data structure.
13. Which are the operations performed on data structure?
14. How to measure the performance of algorithm?
15. What is time complexity?
16. Define space complexity.
17. When empirical testing is used?
18. What is the use of theoretical testing?
19. What is average, best and worst case complexity?
20. Define O notation of time complexity.
21. What is frequency count in apriori analysis?
22. List out the notation that used to express time complexity of algorithm.

Long Questions:-

1. Explain the classification of data structure.
2. Explain various operations performed on data structures.
3. Write a short note on abstract data type.
4. Explain the properties of algorithm.
5. Explain the steps for the development of algorithm.
6. Differentiate between linear and non-linear data structure.
7. What do you mean by algorithm? Give example.
8. Explain efficiency of algorithm.
9. Write a short note on asymptotic notations.
10. Distinguish between best, worst and average case complexities of an algorithm.
11. What do you mean by Time and Space complexity and how to represent these complexity?
12. Explain the concept of data type.
13. Find the complexity of following code.

```
for (int i = 0; i < n; i++)    {
      for(int j=I;j<n;j++)
            cout<<j
                }
```

14. Find the total frequency count of following code.

```
for  send=1 to n do
    for receive =1 to send do
          for ack=2 to receive do
                message=send-(receive+ack)
                ack=ack-1
               send=send+1
         end
 end
```

```
      end
```

15. Find the total frequency count of following code.

```
i=2n
for j=1 to i
for k=3 to j
n=n+1
end
end
```

Fill in the blank:-

1. Logical or mathematical model of particular organization of data is called ______________.
2. Step by step procedure used to solve problem is called ________ _______.
3. ____________ was defined to be set of data object and fundamental operation that can be performed on the set.
4. A __________ refers to the type of values that variable in programming language hold.
5. A list of elements is called as __________.
6. Data structures are classified as _____________ and ___________ data structure.
7. __________ Data structure include array, stack and linked list.
8. _____________ Data structure include tree and graph.
9. The performance of algorithms can be measured on the scale of _________ and space.
10. ___________approach calls for implementing the complete algorithms and executing them on computer for various instances of the problem.
11. ___________approach calls for mathematically determining the resources such as time and space.
12. The disadvantage of __________ testing is that it is dependent on various other factors.
13. The __________ complexity of algorithm can be measured using apriori analysis.
14. _________analysis computes the time complexity as a function of total frequency count of the algorithm.
15. ___, ____, ___ and ____ are asymptotic notations that are used to express the time complexity of algorithm.
16. _____________________ is not just dependent on the input size but is also dependent on the nature of input.
17. Input instance for which algorithm take minimum possible time is called ______________.
18. ___________analysis is appropriate when the response time of the algorithm is critical.
19. _______ serves as upper bound of the performance measured.
20. _______ serves as lower bound of the performance measured.

State True or False:-

1. An algorithm must terminate after an infinite number of steps.
2. An algorithm must be generic enough to solve all problems of a particular class.
3. Algorithms enforce a language or made for its expression but only demand adherence to its properties.
4. There are eight steps perform to development of an algorithm.
5. Algorithm testing is a step to development of an algorithm.
6. Algorithm correctness is a step to development of an algorithm.
7. A data type refers to the type of values that variable in programming language hold.
8. Integer, real, boolean is a non primitive data type.
9. ADT is defined as a set of data objects.
10. ADT focus on what data structure does rather than how it does.
11. Data structures are classified as linear and non-primitive data structure.
12. Array is a linked data structure.
13. Tree is linear data structure.
14. Non linear data structures include only tree data structure.
15. The performance of algorithms can be measured only on the scale of time.
16. Posteriori approach calls for mathematically determining the resources.
17. The time complexity of an algorithm is dependent on parameters associated with the input/output instance

of the problem.

18. The apriori analysis computes the time complexity as a function of the total frequency count of the algorithm.
19. O asymptotic notation serve as the lower bound of the performance measured.
20. Input instance for which algorithm take minimum possible time is called worst case.
21. Input 71,21,9,3,1,5,-23,3,11,33,36,37,-3,-7,11,-5,7,11,-13,17,22 is an example of average case.
22. Input instance for which algorithm take maximum possible time is called best case.
23. Worst case analysis is appropriate when the response time of the algorithm is critical.

Multiple Choice Questions:-

1. Two main measures for the efficiency of an algorithm are
   a. Processor and memory
   b. Complexity and capacity
   c. Time and space
   d. Data and space
2. The time factor when determining the efficiency of algorithm is measured by
   a. Counting microseconds
   b. Counting the number of key operations
   c. Counting the number of statements
   d. Counting the kilobytes of algorithm
3. The space factor when determining the efficiency of algorithm is measured by
   a. Counting the maximum memory needed by the algorithm
   b. Counting the minimum memory needed by the algorithm
   c. Counting the average memory needed by the algorithm
   d. Counting the maximum disk space needed by the algorithm
4. Which of the following case does not exist in complexity theory
   a. Best case
   b. Worst case
   c. Average case
   d. Null case
5. Full form of ADT is
   a. Advanced data type
   b. Array data type
   c. Abstract data type
6. The complexity of the average case of an algorithm is
   a. Much more complicated to analyze than that of worst case
   b. Much simpler to analyze than that of worst case
   c. Sometimes more complicated and some other times simpler than that of worst case
   d. None or above
7. Which of the following data structure is not linear data structure?
   a. Arrays
   b. Linked lists
   c. Both of above
   d. None of above
8. Which of the following data structure is linear data structure?
   a. Trees
   b. Graphs
   c. Arrays
   d. None of above
9. Which of the following data structure is non-linear type?
   a. String
   b. Lists
   c. Stacks

d. None of above

10. Which of the following data structure is linear type?

a. Strings
b. Lists
c. Queues
d. All of above

11. _____________is a step-by-step procedure for calculation

a. Data structure
b. Abstract Data Type
c. Primitive Data Type
d. Algorithm

12. A mathematical-model with a collection of operations defined on that model is called

e. Data structure
f. Abstract Data Type
g. Primitive Data Type
h. Algorithm

13. Representation of data structure in memory is known as:

a. Recursive
b. Abstract data type
c. Storage structure
d. File structure

14. Which of the following is considered an Abstract Data Type?

a. Array
b. reference variable
c. any of the primitive types (e.g., int, double, char)
d. Stack
e. all of the above

15. An algorithm must be generic enough to solve all problems of a particular class.

a. Finiteness
b. Definiteness
c. Generality
d. Effectiveness

16. The first step of development of an algorithm is

a. Problem analysis
b. Problem statement
c. Algorithm analysis
d. Implementation

17. Input instance for which algorithm take minimum possible time is called

a. Worst case
b. Best case
c. Average case
d. Null case

18. Input instance for which algorithm take maximum possible time is called

a. Worst case
b. Best case
c. Average case
d. Null case

19. Which case analysis appropriate when the response time of the algorithm is critical?

a. Worst case
b. Best case
c. Average case

d. Null case

20. The time complexity of the algorithm in a best case would be expressed as
    a. O(1)
    b. O(n)
    c. O(n2)
    d. O(n+1)

Unit : 2 Arrays

Short Questions:-

1. Define array.
2. List out application of Array.
3. Which operation is supported by an array ADT?
4. What will happen in a C++ program when you assign a value to an array element whose subscripts exceed the size of array?
5. What is the index number of the last element of an array with 20 elements?
6. List out the operations performed on Array.
7. Give the number of elements in array a[1:5].
8. Give the number of elements in array a[1:5,1:4,1:3].
9. Give the number of elements in array A[3][2].
10. What is row major order?
11. What is column major order?
12. Write formula to calculate address of elements in one-dimensional array.
13. Write formula to calculate address of elements in two-dimensional array.
14. Write formula to calculate address of elements in three-dimensional array.
15. Define sparse matrix.
16. Define order-list matrix.
17. If the starting address of array a[-2,23] is 100 then what will be the address of 16th element?
18. If the starting address of array a[1:5,1:6] is 100 then what will be the address of a[3,4] element?
19. If the starting address of array a[1:5,1:6,1:4] is 100 then what will be the address of a[3,4,5] element?
20. Write any one difference between row major and column major.
21. What are the disadvantages of array?

Long Questions:-

1. What is an array? Which operations can be performed on Array? Explain with example.
2. How to calculate number of elements in one dimensional array? Explain with example.
3. How to calculate number of elements in two dimensional arrays? Explain with example.
4. How to calculate number of elements in three dimensional arrays? Explain with example.
5. Explain one-dimensional array. How one dimensional array can be represented in memory?
6. Explain two-dimensional array. How two dimensional arrays can be represented in memory?
7. Explain three-dimensional array. How three dimensional arrays can be represented in memory?
8. Explain any one method to calculate memory location for different position in two-dimensional array.
9. What are the applications of an array? Explain each with examples.
10. Explain sparse matrix. What are the benefits of the sparse matrix?
11. Explain order-list matrix. What are the benefits of the order-list matrix?
12. Write an algorithm for insert and delete operation in array.
13. Write an algorithm to implement sparse matrix.
14. Write an algorithm to search element in array.
15. Write program to insert element at position of user choice.
16. For the following array A, compute
    a. the dimension of A
    b. the space occupied by A in the memory
    c. the address of A[7,2]

Array: A Column Index: 0:5

Base address: 100 Size of memory location: 4 bytes
Row Index: 0:15

17. Distinguish between the row major and column major ordering of an array.
18. Suppose A is linear array with n numeric values. Write procedure which finds the average of the values in A.
19. Write a program to find second highest value from array elements.
20. Write a program to delete an element of array at position of user choice.

Fill in the blanks:-

1. ________________ was defined to be a set of data objects and operations that ca be performed on this set.
2. _____________ is an ADT whose objects are sequence of elements of the same type.
3. ___________and _________________ are two basic operations performed on array.
4. One-dimensional arrays are mathematically linked to ______________________________.
5. Two-dimensional arrays are mathematically linked to ______________________________.
6. Three-dimensional arrays are mathematically linked to ______________________________.
7. In the case of the array A[l:u] where l is _____________ and u is a _____________ of the index range.
8. The array A[5:30] will have ________ number of elements.
9. The array A[1:5,1:3] will have ________ number of elements.
10. The array A[1:3, 3:4, 2:6] will have ________ number of elements.
11. __________and ________________ are the two ways to store array in memory.
12. The starting index of array is ______.
13. The starting memory location of an array is called _______________.
14. ________________ will be the address of a 5th element of an array having base address 100.
15. If the array is A[1:10, 1:5], _____________ will be the address of A[5,3] element of an array having base address 100.
16. If the array is A[-2:4, -6:10, 1:3], _____________ will be the address of A[-1,-3,2] element of an array having base address 100.
17. The matrix with zeros as its dominating elements is called _________________.
18. ____________and ____________________ are the applications of array.
19. ____________matrix is a matrix with zero as the dominating elements.
20. The elements of the list are known as _________________.

State True or False:-

1. Array is a linear Data structure.
2. Array is abstract data type.
3. Array is a Non primitive Data type.
4. The elements of the array are stored continuously in memory location.
5. Array as an ADT supports only two operations STORE and RETRIVE.
6. If a is an array the operations can be represented as STORE(a,i).
7. If a is an array the operations can be represented as RETRIVE(a,i,e).
8. The A[1] define first element in array in C++.
9. Array can be store in either row or column major.
10. The type of all elements in an array must be same.
11. When we declare an array in C++, it is automatically initialized its elements to zero.
12. The array int A[10]; can store 11 elements.
13. (u-l-1) formula is used to calculate number of elements in one dimensional array
14. (u1-l1+1)(u2-l2+1) is used to calculate number of elements in two dimensional array
15. Accessing an array outside its range is a compile time error.
16. In multidimensional array, it is not necessary to give all dimension of the array.
17. A char type variable cannot be used in array.
18. In C++, we can use maximum of 3 dimensions for an array.
19. An order list matrix can be either empty or non empty.
20. Recursion is an application of array.

Multiple Choice Questions:-

1. Which of the following linear Data Structure?

a. Array b. Structure c. Tree d. Long

2. Consider following code:

```
 #include<iostream.h>
 int main()
 {     int a[10];
        A[0]=10; A[1]=4; A[3]=15;
       cout<<A[2+1];
 }
```

What will be the output of following code?

a. NULL b. 10 c. 4 d. 15

3. Which are the correct array initialization statements?
   a. int A[3]={1,2,3};
   b. int A[3]={123};
   c. int A[3]=”123”;
   d. All
4. Which of the following statements are wrong statements?
   a. Array is primitive data structure.
   b. Every element of array must be same.
   c. In array, Insert element is called push operation.
   d. All
5. Which are the applications of array?
   a. Sparse matrix
   b. Ordered list
   c. Both a & b
   d. none
6. Which among the following pairs of operations is supported by an array ADT?
   a. Store and Retrieve
   b. Insert and Delete
   c. Copy and Delete
   d. Append and Copy
7. The number of elements in array Array[1:u] is given by
   a. (1 - u)
   b. (u)
   c. (u – 1 + 1)
   d. (u – 1 - 1)
8. The number of elements in array Array[l1:u1, l2:u2] is given by
   a. (u1- l1 – 1)(u2 – l2 - 1)
   b. (u1*u2)
   c. (u1 – l1)(u2 – l2)
   d. (u1- l1 + 1)(u2 – l2 + 1)
9. The number of elements in array Array[l1:u1, l2:u2, l3:u3] is given by
   a. (u1- l1 – 1)(u2 – l2 - 1) (u3 – l3 - 1)
   b. (u1*u2*u3)
   c. (u1 – l1)(u2 – l2)(u3-l3)
   d. (u1- l1 + 1)(u2 – l2 + 1)(u3 – l3 + 1)
10. For the array A[1:u1, 1:u2] where α is the base address, A[i,1] has its address given by
    a. (i – 1)u2
    b. α + (i – 1)u2
    c. α + i * u2
    d. α + (I – 1) * u1
11. An one-dimensional array array[1:5] contains _________ elements.

    a. 5 b. 4 c. 1 d. 6

12. A two-dimensional array array[1:3, 1:3] contains _________ elements.
    a. 3 b. 6 c. 9 d. 7
13. A multi-dimensional array array[0:2, 10:20, 3:4, -10:2] contains _________ elements.
    a. 240 b. 858 c. 390 d. 160
14. The memory address of the first element of an array is called
    a. floor address
    b. foundation address
    c. first address
    d. base address
15. The memory address of fifth element of an array can be calculated by the formula
    a. LOC(Array[5]=Base(Array)+w(5-lower bound), where w is the number of words per memory cell for the array
    b. LOC(Array[5])=Base(Array[5])+(5-lower bound), where w is the number of words per memory cell for the array
    c. LOC(Array[5])=Base(Array[4])+(5-Upper bound), where w is the number of words per memory cell for the array
    d. None of above
16. Which of the following data structures are indexed structures?
    a. linear arrays
    b. linked lists
    c. both of above
    d. none of above
17. Two dimensional arrays are also called
    a. tables arrays
    b. matrix arrays
    c. both of above
    d. none of above
18. If the array is A[1:10], what will be the address of A[5] element of an array having base address 100.
    a. 105
    b. 106
    c. 107
    d. 108
19. If the array is A[1:10, 1:5], ____________ will be the address of A[5,2] element of an array having base address 100.
    a. 120
    b. 121
    c. 122
    d. 123
20. If the array is A[1:5, 1:2, 1:3], ____________ will be the address of A[1,2,3] element of an array having base address 100.
    a. 102
    b. 104
    c. 105
    d. None

Unit-3 Stack and Queue

Short Questions:-

1. Define Stack.
2. Give real world example of stack.
3. List the operations on stack.
4. List the application on stack.
5. Define push operation on stack.

6. Define pop operation on stack.
7. Define peep operation on stack.
8. When stack is said to be overflow?
9. Give definition of infix, prefix and postfix notation.
10. Define Tail recursion.
11. Identify the types of expression whether it is infix, prefix or postfix.
    a. 4,2$3*3-8,4/1,1+/+
    b. PQ+R+-S↑UV+*
12. Define Queue.
13. Give real world example of Queue.
14. List the operations on Queue.
15. List the application on Queue.
16. Define Insertion operation on Queue.
17. Define Deletion operation on Queue.
18. Define Circular Queue.
19. List out limitation of linear queue.
20. Write algorithm to insert element into circular queue.
21. What is Deques? Explain with example.
22. Define priority queue.

Long Questions:-

1. Explain Stack with its example.
2. Explain the operation performed on Stack.
3. Explain Push operation with algorithm.
4. Explain Pop operation with algorithm.
5. Explain Peep operation with algorithm.
6. Explain application of Stack.
7. Explain Evaluation of expressions on stack.
8. Write pseudo-code for factorial computation.
9. Evaluate following expression.
    a. 10+3-2-8/2*6-7
    b. (12-(2-3)+10/2+4*2)
10. Convert following infix expression to postfix expression:
    a. ((a+b)/d-((e-f)+g)
    b. 12/3*6+6-6+8/2
11. Convert following infix expression to prefix expression:
    a. ((a+b)/d-((e-f)+g)
    b. 12/3*6+6-6+8/2
12. Convert following postfix expression to infix expression:
    c. 4,2$3*3-8,4/1,1+/+
    d. PQ+R+-S↑UV+*
13. Convert following postfix expression to prefix expression:
    e. 4,2$3*3-8,4/1,1+/+
    f. PQ+R+-S↑UV+*
14. Explain Queue with its example.
15. Explain the operation performed on Queue.
16. Explain insertion operation for queue with algorithm.
17. Explain Deletion operation for queue with algorithm.
18. Explain application of Queue.
19. Write short note on Deque.
20. Write short note on Priority Queue.
21. What are the difference between stack and queue.
22. How to overcome limitation of linear queue? Explain in detail.

Fill in the blank:-

1. A _______________ data structure is an ordered list with insertion and deletion done at one end of the list known as top of stack.
2. Stack can be stored in _________ and ______________.
3. An insertion operation is called as ____________.
4. A deletion operation is called as ___________.
5. If top pointer's value is equal to the size of the stack then Stack is ___________.
6. When a function definition includes a call itself, it is referred to as a ___________.
7. ______________operation is called as push and ___________operation is called as pop operation.
8. ______________ is a special case of recursion where a recursive call to function turns out to be the last action in the calling function.
9. In recursive call to store the value of the variables and starting address execution is maintained in ____________.
10. Polish notation classified into ____________, _____________ and ______________.
11. When the operators are written before their operands then the resulting expression is called _________ polish notation.
12. When the operators exist between two operands then the expression is called ___________ expression.
13. When the operators come after their operands then the resulting expression is called _________ polish notation.
14. A+B (C/D) is an example of _____________ expression.
15. +A*BC is an example of _____________ expression.
16. Z+YX*WVU/T*-S*+ is an example of _____________ expression.
17. A _____________ data structure is linear list in which all insertion are made at the rear end of list and deletion are made at the front of end of list.
18. Queue data structure support ____________ and ____________ operations.
19. For a queue implemented as an array, the initial value of the front and read is set to ________.
20. A ____________ is a queue structure in which elements are inserted or deleted based on priority.
21. A _____________ with insertion and deletion done at either ends or may be appropriate restricted at one of the ends.
22. A ______________ queue has been demonstrated on the problem of job scheduling in time-sharing system environment.
23. ____________is an application of priority queue.

State True or False:-

1. Stack is not an ADT.
2. Stack is non linear data structure.
3. Police notation is an application of stack.
4. Stack is also known as FIFO Data Structure.
5. Queue is also known as LIFO Data Structure.
6. STACK_FULL when elements whose number is over and n are attempted to be push into the stack.
7. <operand><operator><operand> is known as Infix expression.
8. <operand><operator><operator> is known as postfix expression.
9. <operator><operand><operand> is known as prefix expression.
10. Recursive programming is application of Stack.
11. In Queue insertion and deletion is done at one end called top.
12. Insertion is done on one end called front.
13. Space utilization is good in Circular queue as compare to Simple queue.
14. Front and Rear two pointers are maintained in Queue.
15. Insertion operation is also referred to as enqueuing.
16. Deletion operation is also referred to as dequeuing.
17. Space utilization is good in circular queue as compare to simple queue.
18. Tower of Hanoi is an application of Queue.
19. Traffic control system is an application of Queue.

20. Circular queue and Circular Link list are same.

Multiple Choice Questions:-

1. The condition Top=-1 indicates that
   a. Stack is empty
   b. Stack is full
   c. Stack has only one element
   d. None of these
2. Which of the following name related to stacks?
   a. FIFO
   b. PUSH
   c. POP
   d. ALL
3. Example of primitive recursion is
   a. Tower of Hanoi
   b. Ackermann’s function
   c. Both
   d. None
4. Stack works on the principles:
   a. FCFS
   b. LIFO
   c. Both a &b
   d. None
5. The term push and pop is related to the
   a. Array
   b. Lists
   c. Stacks
   d. All of above
6. Which of the following is the condition of circular queue overflow?
   a. Front=0 and Rear=size
   b. Front+1=Rear
   c. Both a & b
   d. Neither a nor b
7. A data structure where elements can be added or removed at either end but not in the middle
   a. Linked lists
   b. Stacks
   c. Queues
   d. Deque
8. In which notation operator is comes between operand?
   a. Infix
   b. Prefix
   c. Postfix
   d. None
9. In which notation operator is comes after operand?
   a. Infix
   b. Prefix
   c. Postfix
   d. None
10. In which notation operator is comes before operand?
   a. Infix
   b. Prefix
   c. Postfix

d. None

11. Which of the following is not a application of Stack?
    a. Evaluation of Police notation
    b. Tower of Hanoi
    c. Stack Machine
    d. None
12. Queue works on the principles:
    e. FCFS
    f. LIFO
    g. Both a &b
    h. None
13. Which of the following is related to Queue?
    a. Round Robin algorithm
    b. Trafic Control System
    c. All
    d. None
14. The infix expression for the postfix expression : 5,6,2+*12,4/-
    a. 5*(6+2)-12/4
    b. 5+6-2*12/4
    c. (5+6)-2/12*4
    d. None of above
15. Answer of following postfix expression: 2,3,10+*8,2/-
    a. 20
    b. 22
    c. 23
    d. 24
16. The postfix expression for the infix expression : a+b*c/d
    a. abc*d/+
    b. a*bcd/+
    c. ab*cd/+
    d. abcd*/+
17. The prefix expression for the infix expression : a+b*c/d
    a. +ab*/cd
    b. +*ab/cd
    c. +a*b/cd
    d. None
18. Which of the following is not a type of Dequeue?
    a. Input Restricted Queue
    b. Output Restricted Queue
    c. a & b both
    d. None
19. Which data structure will you use to evaluate prefix notation?
    a. Queue
    b. Stack
    c. Linked List
    d. Array
20. Which of the following is not the operation on stack?
    a. Push
    b. Pop
    c. Peep
    d. Enqueue
21. Which of the following is not the operation on Queue?

a. Insertion
b. Deletion
c. Updating

Unit-4 Linked List

Short Questions:-

1. What is the limitation of sequential data structures?
2. What is linked list?
3. Give real world example of linked list.
4. Explain logical representation of linked list.
5. What are the advantages of singly linked list?
6. What are the disadvantages of singly linked list?
7. Which are the operations performed in singly linked list?
8. What is the need for linked representation of lists?
9. Define circular linked list.
10. What are the advantages of circular linked list?
11. What are the disadvantages of circular linked list?
12. What is the node structure for circular linked list?
13. Define doubly linked list.
14. What are the advantages of doubly linked list?
15. What are the disadvantages of doubly linked list?
16. List out operations performed in doubly linked list.
17. List application of linked list.
18. What is the difference between circular linked list and linear linked list?
19. What is the difference between array and stack?
20. What do you mean by polynomials?
21. Give node structure for the term of polynomial having single variable.
22. How singly linked list representation of polynomials?
23. Define sparse matrix?

Long Questions:-

1. Write short note on linked list.
2. Explain operation of singly linked list with algorithm.
3. Explain circular linked list.
4. What are the advantages of circular linked list over singly linked list?
5. Write pseudo code to add node at the end in circular linked list.
6. Explain doubly linked list with advantage and disadvantage of it.
7. Write a pseudo code to delete a node from doubly linked list.
8. Explain operation of doubly linked list with algorithm.
9. Write short note on multiply linked lists.
10. Explain application of linked list.
11. Write short note on polynomial manipulation.
12. Write short note on sparse matrix.
13. Explain operation of linked stack and linked queue.
14. Write algorithm for push/pop operation on a linked stack.
15. What are merit of linked stack and queues over their sequential counterparts?
16. How are push and pop operations implemented on a linked stack?
17. Write algorithm for insertion/deletion operation on a linked queue.
18. Write short note on Dynamic memory management.
19. Explain application of linked stack and linked queue.
20. Write a pseudo code for implementing stack using linked list.
21. Write a pseudo code for implementing queue using linked queue.

Fill in the blank:-

1. A linked representation of data structure known as ______________ is a collection of ________.
2. The operation and management of linked data structure are less prone to create ____________.
3. A ____________ is linear data structure, each node of which has one or more data items field but only a single link field.
4. Node is collection of ________.
5. Linked field are also referred as ____________.
6. If START pointer is not null then singly linked list is ____________.
7. Singly linked list has its last node carrying a _______ pointer.
8. In _______ linked list, nodes are circularly linked.
9. A ____________ is a linked linear data structure, each node of which has one or more data fields but only left and right link.
10. A doubly linked list may or may not have ____________ node.
11. The availability of two links __________ and _________ permit forward and backward movement during the processing of the list.
12. _____________ is an application of singly linked list.
13. A ____________ linked list in its simplest form may represent a cluster of singly linked list network together.
14. _____________ is an application of multiply linked list.
15. A ___________ is also a linear list of elements commonly implemented as linked list with two pointers.
16. Dynamic memory management deal with method of _______________ and _____________ for future use.
17. The automatic recycling of dynamic allocated memory is known as ________________.
18. Free storage pool is also referred as _____________.
19. ___________ is an application of linked Stack.
20. ___________ is an application of linked Queue.

State True or False:-

1. Linked list is linear data structure.
2. Linked list is a sequential data structure.
3. Linked list is used to store fixed size of data.
4. Drawback of linked list is efficient to implementation of insertion and deletion operation.
5. Linked list is collection of nodes.
6. The empty link field is also referred to as null link.
7. If START pointer is not null then singly linked list is empty.
8. Singly linked list has its last node carrying a null pointer.
9. A doubly linked list may or may not have head node.
10. There is no null linked in singly linked list.
11. There is no null linked in doubly linked list.
12. There is no null linked in circular linked list.
13. Sparse matrix is application of linked stack.
14. Infinite loop problem can be arise in doubly linked list.
15. Polynomial manipulation is application of linked list.
16. In a linked stack, the top pointer represent first node.
17. The time complexity of push operation in a linked stack is O[1].
18. The time complexity of deletion operation is O(1).
19. Free storage pool is also referred to as Available space.
20. ((A+B)↑C-D)+E-F is a example of imbalanced arithmetic expression.

Multiple Choice Questions:-

1. A node carries information regarding
    a. Data
    b. Link
    c. Link and Data
    d. None

2. A linked list is which type of data structure.
   a. Linear
   b. Non Linear
   c. Hierarchical
   d. None
3. Type of storage is used to represent Lists
   a. Random
   b. Sequential
   c. Dynamic
   d. Logical
4. Linked list are best suited
   a. For relatively permanent collections of data
   b. For the size of structure is constantly changed
   c. Both a & b
   d. None
5. Linear order linked list is provided through _________
   a. variables
   b. arrays
   c. Pointer
   d. Strings
6. In a Single Link List_________ node contains no links.
   a. First
   b. Last
   c. last but one
   d. middle
7. In Single Linked List a node contain minimum how many fields(assuming one for data).
   a. 2
   b. 3
   c. 1
   d. None
8. Single link list performs which of the following methods
   1) Insertion
   2) Modification
   3) Searching
   a. 1 and 2
   b. 2 and 3
   c. 1 and 3
   d. All
9. In linked lists there are no NULL links in:
   a. Singly linked list
   b. Doubly linked list
   c. Circular linked list
   d. None
10. The list with no node is called as
    a. Empty list
    b. Null list
    c. Zero list
    d. None
11. Which of the following is the application of the singly linked list?
    a. Sparse matrix
    b. Police notation
    c. Tower of Hanoi

d. All

12. Which of the following will contain more memory space?
    a. Singly linked list
    b. Doubly linked list
    c. Array
    d. Circular linked list
13. In polynomial manipulation, nodes consists of three field representing
    a. Coefficient, exponential and link
    b. Previous item link, data item, next item link
    c. Coefficient, data item and link
    d. None
14. A linked list in which last node contain the link of the first node is called
    a. Singly linked list
    b. Doubly linked list
    c. Circular linked list
    d. All
15. A singly linked list facilitates list traversal in
    a. Single direction
    b. Any direction
    c. Circular direction
    d. All
16. A doubly linked list facilitates list traversal in
    e. Single direction
    f. Any direction
    g. Circular direction
    h. All
17. Linked list START=NULL is ____________
    a. Underflow
    b. Overflow
    c. None
18. In a linked list, the pointer of the last node contains the special value called _______________ linked.
    a. Linked to the first node
    b. Null
    c. Link
    d. Pointer to the tail node
19. Which of the following is linear data structure?
    a. Tree
    b. Graph
    c. Linked List
    d. All
20. In which linked list, nodes in form of ring?
    a. Singly linked list
    b. Doubly linked list
    c. Circular linked list
    d. All
21. Balancing symbol is a application of _______________.
    a. Singly linked list
    b. Doubly linked list
    c. Circular linked list
    d. Linked stack
22. What kind of list is best to answer questions such as "What is the item at position n?"
    a. Lists implemented with an array

b. Doubly-linked lists
c. Singly-linked lists.
d. Doubly-linked or singly-linked lists are equally best

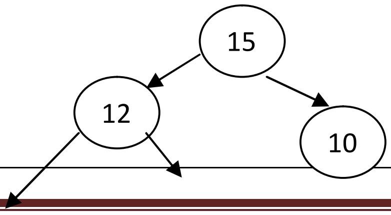
Unit-5 Trees and Binary Trees

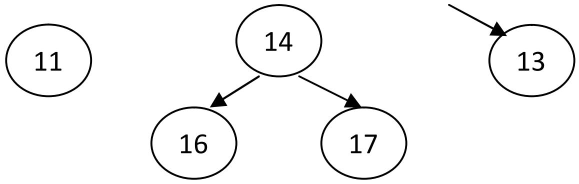
Short Questions:-

1. Define tree.
2. What is degree of node?
3. Define sibling.
4. Define forest. Also give example of it.
5. Define binary tree.
6. List out type's binary tree.
7. What is the difference between full binary tree & complete binary tree?
8. List out different techniques to represent tree.
9. List out different operations you can perform on tree.
10. List out traversal of binary tree.
11. What is inorder traversal?
12. What is preorder traversal?
13. What is postorder traversal?
14. What is the maximum number of nodes in a binary tree of depth k?
15. Trace the binary tree of inorder traversal: BFGPRSTWYZ.
16. What are the applications of tree?
17. Trace the binary tree of preorder traversal: PFBHGSRYTWZ.
18. What do you mean by expression tree?
19. Define leaf node and siblings with example.
20. What is threaded binary tree?

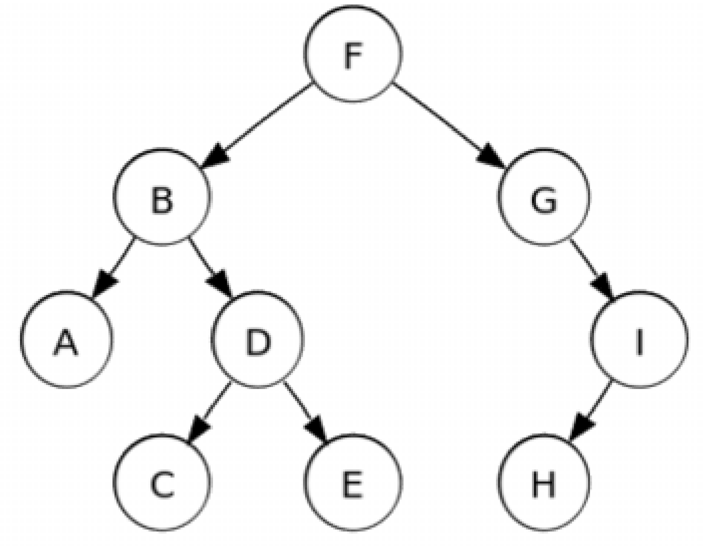
Long Questions:-

1. Explain tree data structure.
2. How to represent tree using linked list?
3. Explain binary tree with its representation including advantage and disadvantage.
4. Write a code to insert a node in a binary tree.
5. Write a code to delete a node in binary tree.
6. Explain array representation of binary tree with example?
7. Explain linked representation of binary tree with example?
8. Explain traversal technique of binary tree.
9. Explain inorder traversal with example.
10. Explain preorder traversal with example.
11. Explain postorder traversal with example.
12. Explain application of binary tree.
13. Create a binary tree using inorder and preorder traversal
    Inorder: D B H E A I F J C G, Preorder: A B D E H C F I J G
14. Create a binary tree using inorder and postorder traversal
    Inorder: D B H E A I F J C G, Postorder: D H E B I J F G C A
15. Create a binary tree from the following sequence:
    14, 34, 22, 44, 11, 24, 33
16. Using the following binary tree traverse it into inorder, preorder and postorder:





17. Using the following binary tree traverse it into inorder, preorder and postorder:



18. Consider the following tree.

```
          A
        /   \
       B     C
      / \   / \
     D  E  H   I
    /      /\
   F     J   K
              \
               L
```

a. How many leaves does it have?
b. How many of the nodes have at least one sibling?
c. What is the value stored in the parent node of the node containing 30?
d. How many descendants does the root have?
e. What is the depth of the tree?
f. How many children does the root have?

19. What is inorder traversal of binary tree? Write inorder traversal of given binary tree.

```
          A
         /\
        B   C
       /     \
      D       E
     /\
    G  F
```

20. Write algorithm to perform inorder traversal of binary tree.

21. Write algorithm to perform preorder traversal of binary tree.
22. Write algorithm to perform postorder traversal of binary tree.

Fill in the blank:-

1. _______________and ______________ are non linear data structures.
2. Links between two nodes term as ________________.
3. The number of subtrees of a node node is known as ______________.
4. These nodes which hang from branches emanating from a node are known a ______________ and the node from which the branches emanate is known as __________ node.
5. Children of the same parent node are referred to as ______________.
6. The ____________________ is the maximum degree of the node in the tree.
7. The __________ of a tree is defined to be the maximum level of any node in the tree.
8. A ________________ is a set of zero or more disjoint tree.
9. _____________ has the characteristic of all nodes having at most two branches.
10. A binary tree of height which as all its permissible maximum number of nodes is known as ____________________.
11. A binary tree could be represented using a _______________data structure as well as ______________ data structure.
12. An important operation that is performed on a binary tree is known as ___________.
13. A traversal is governed by three action ____________, ______________ and ______________.
14. A traversal keep moving left in the binary tree until one can move, process the node and moves to the right to continue is called as ___________ traversal.
15. Left, root, right traversal known as ___________ traversal.
16. Root, left, right traversal known as ___________ traversal.
17. Left, right, root traversal known as ___________ traversal.
18. In _______________ algorithm, the deletion procedure is complex.
19. ______________ is the process of visiting every node in a tree at least once.
20. In ____________traversal, the root node is visited last.
21. Children of the same parent are called ________________.
22. Nodes which are subtrees of another node are called ________________.
23. If a node is a terminal node, then its left child and right child field are filled with _____________.

State True or False:-

1. Tree is a non linear data structure.
2. To represent hierarchical relationship between elements, tree data structure is used.
3. Linked between node terms as tree.
4. The specially designated node is called root.
5. In a tree diagram, a circle represents nodes.
6. A tree can represent many-to-many relationships.
7. The number of subtrees of a node is known as degree of the node.
8. Nodes that have zero degree are known as non terminal nodes.
9. Children of the same parent node are referred to as sibling.
10. A node of n children should have n values.
11. A forest is a set of zero or more disjoint trees.
12. A binary tree has the characteristic of all nodes having at most three branches.
13. A binary tree which is dominated solely by left child node is called full binary tree.
14. Array representation ideally suits a full binary tree due to its non wastage space.
15. The tree is accessed by remembering the pointer to the root of the tree.
16. In binary tree algorithm, the deletion procedure is complex.
17. Traversal is the process of visiting every node in a tree at least once.
18. In preorder traversal, the root node is visited first.
19. In preorder traversal, the root node is visited last.
20. All nodes in a list point to some other node.

21. In a binary tree, a node may have a degree greater than 2.

Multiple Choice Questions:-

1. Which of the following data structure is non-linear type?
   a. Strings
   b. Lists
   c. Stacks
   d. Tree
2. To represent hierarchical relationship between elements, which data structure is suitable?
   a. Deque
   b. Priority
   c. Tree
   d. All of above
3. A binary tree whose every node has either zero or two children is called
   a. Complete binary tree
   b. Binary search tree
   c. Extended binary tree
   d. None of above
4. The depth of a complete binary tree is given by
   a. $D_n = n \log_2 n$
   b. $D_n = n \log_2 n+1$
   c. $D_n = \log_2 n$
   d. $D_n = \log_2 n+1$
5. The post order traversal of a binary tree is DEBFCA. Find out the in order traversal
   a. ABFCDE
   b. ADBFEC
   c. ABDECF
   d. None
6. The post order traversal of a binary tree is DEBFCA. Find out the pre order traversal
   a. ABFCDE
   b. ADBFEC
   c. ABDECF
   d. ABDCEF
7. The in order traversal of a binary tree is ABFCD. Find out the pre order traversal
   a. ABFCD
   b. ADBFC
   c. ABDCF
   d. None
8. The in order traversal of tree will yield a sorted listing of elements of tree in
   a. Binary trees
   b. Binary search trees
   c. Heaps
   d. None of above
9. Consider the following tree.

```
      14
     / \
    2   11
   /\  /  \
  1  3  10  30
     /  /
    7  40
```

   a. How many leaves does it have?

        a. 2
        b. 4
        c. 6
        d. 8
        e. 9
    b. How many of the nodes have at least one sibling?
        a. 5
        b. 6
        c. 7
        d. 8
        e. 9
    c. What is the value stored in the parent node of the node containing 30?
        a. 10
        b. 11
        c. 14
        d. none
    d. How many descendants does the root have?
        a. 0
        b. 2
        c. 4
        d. 8
    e. What is the depth of the tree?
        a. 2
        b. 4
        c. 8
        d. None
    f. How many children does the root have?
        a. 6
        b. 7
        c. 8
        d. 9

10. Which of the following statement is false?
    a. Every tree is a bipartite graph
    b. A tree contains cycle
    c. A tree with n nodes contains n-1 edges
    d. A tree is connected graph
11. A full binary tree with n leaves contains
    a. n nodes
    b. $\log_2 n$ nodes
    c. 2n – 1 nodes
    d. 2n + 1 nodes
12. A full binary tree with n non-leaf nodes contains
    a. $\log_n$ nodes
    b. n + 1 nodes
    c. 2n – 1 nodes
    d. 2n + 1 nodes
13. A complete binary tree of level 5 has how many nodes?
    a. 15
    b. 25
    c. 63
    d. 30
14. Traversing a binary tree first root and then left and right subtrees called _____________ traversal.

a. Postorder
b. Preorder
c. Inorder
d. None of these

15. A binary tree of depth "d" is an almost complete binary tree if
a. Each leaf in the tree is either at level
b. For any node
c. Both a & b
d. None of these

16. The maximum number of nodes on level i of a binary tree is
a. 2i.1
b. 3i.1
c. I + 1
d. 2i + 2

17. Number of possible binary trees with 3 nodes is
a. 12
b. 9
c. 14
d. 15

18. A binary tree is generated by inserting in order of the following integers: 50, 15, 62, 5, 20, 58, 3, 8, 37, 60, 24 the number of nodes in the left of the root respectively is
a. (4,7)
b. (7,4)
c. (6,3)
d. (3,6)

19. Consider the following tree

```
          6
         / \
        4    12
       / \   / \
      1   5  10 30
               \
                11
```

If this tree is used for sorting, then a new number 8 should be placed as the
a. Left child of the node labeled 30
b. Right child of the node labeled 5
c. Right child of the node labeled 30
d. Left child of the node labeled 10

20. Which of the following traversal techniques lists the nodes of a binary search tree in ascending order?
a. Post-order
b. In-order
c. Pre-order
d. None of the above

Unit-6 Searching and Sorting

Short Questions:-

1. Define ordered linear search.
2. Define unordered linear search.
3. Give any one difference between order linear searches and unordered linear search.
4. Write down complexity of worst case and best case in ordered linear search.
5. Write down complexity of worst case and best case in unordered linear search.

6. Define binary search.
7. What are the difference between linear search and binary search?
8. In linked list, which searching technique is better linear search or binary search?
9. What are the advantages of binary search over linear search?
10. Define sort.
11. Write down complexity of bubble sort and in which situation bubble sort should be used?
12. Write down complexity of insertion sort and in which situation bubble sort should be used?
13. Write down complexity of selection sort and in which situation bubble sort should be used?
14. Write down complexity of quick sort and in which situation bubble sort should be used?
15. Write down complexity of merge sort and in which situation bubble sort should be used?
16. What is the time complexity of merge sort?
17. What is the time complexity of quick sort?
18. Can bubble sort ever perform better than quick sort?
19. Which sorting techniques are an example of divide and conquer?
20. Which sorting techniques is an application of recursion?

Long Questions:-

1. Write short note on linear search with algorithm.
2. Write short note on binary search with algorithm.
3. What are the merits and demerits of binary search?
4. Write the pseudo-code for linear search using array and linked list.
5. Write pseudo-code for binary search using array and linked list.
6. Explain decision tree for binary search.
7. Write short note on bubble sort with algorithm.
8. Write short note on insertion sort with algorithm.
9. Write short note on selection sort with algorithm.
10. Write short note on merge sort with algorithm.
11. Write short note on quick sort with algorithm.
12. Which sorting technique is more suitable to use? Explain in detail.
13. Write pseudo-code for binary search using array and linked list.
14. Write down procedure for bubble sort.
15. Why is bubble sort stable?
16. Write down procedure for insertion sort.
17. Write down procedure for selection sort.
18. Write down procedure for merge sort.
19. Write down procedure for quick sort.
20. Trace quick sort on the list L= {11, 34, 67, 78, 78, 78, 99}. What are your observations?

Fill in the blank:-

1. ________________search is one where a key K is searched for, in a linear list L of data elements.
2. Ordered linear search reports a time complexity of worst case is _________ and base case is _________ in terms of key comparisons.
3. Unordered linear search reports a time complexity of worst case is _________ and base case is _________ in terms of key comparisons.
4. The complexity of linear search algorithm is ________________.
5. ________________is an algorithm design technique where to solve a problem; the problem is divided in to sub-problems.
6. In the case of binary search, the _____________________ aspect of the technique breaks the list into two sub lists.
7. The complexity of binary search algorithm is ________________.
8. A sorting technique is said to be __________if keys that are equal retain their relative orders of occurrence even after sorting.
9. If the lists of data to be sorted are small enough to be accommodated in the internal memory of the computer, then it is known as ____________.

10. If the lists of data to be sorted are small enough to be accommodated in external storage device, then it is known as ____________.
11. The complexity of Bubble sort algorithm is _______________.
12. The complexity of insertion sort algorithm is _______________.
13. The average case performance of insertion sort reports ________complexity.
14. __________and __________sort is not stable.
15. The complexity of selection sort algorithm is _______________.
16. ____________ is a process by which two ordered list of elements are combined into single ordered list.
17. The complexity of merge sort algorithm is _______________.
18. ________________merge sort procedure is built on the design principle of divided and conquers.
19. The two-way merge sort principle could be extended to k ordered lists in which case it is ermed as ___________ merging.
20. ________sort procedure formulated by C.A.R.
21. The complexity of quick sort algorithm is _______________.

State True or False:-

1. The Sequential Search method on sorted lists is faster than the indexed method.
2. The search technique which loads only a part of the database into main memory is known as external search.
3. A Binary Search can only be applied to sorted records.
4. In Binary Search, when the key is less than the middle element in a sorted array, the higher limit is modified for the next iteration.
5. Binary Search is the fasted of all methods for sorted records.
6. The lower limit is modified when the key is greater than the middle element in the array in a binary search method.
7. Sorting is always performed on the elements stored in primary memory.
8. Minimal storage sorts are optimal for arrays having a large number of elements.
9. The process of sorting a list stored in a file in secondary memory is known as internal sorting.
10. Methods that are not Data Sensitive require the same time to sort an array.
11. The sort is performed according to the key value of each record.
12. Bubble Sort is so named because it bubbles the smallest element to the middle of the array.
13. Bubble sort is instable sort.
14. The complexity of insertion sort is O(n2)
15. Selection sort is not stable.
16. The Quick Sort Algorithm works by partitioning the array to be sorted, then recursively sorting each partition.
17. The insertion sort method is optimal because the sorted array is developed without using any extra storage space.
18. Merge sort is not stable.
19. Quick sort procedure formulated by C.A.R.
20. Quick sort reports a worst case performance when the list is already in its ascending order.
21. The worst case time complexity of the algorithm is O(n).

Multiple Choice Questions:-

1. What is the worst-case time for serial search finding a single item in an array?
   a. Constant time
   b. Logarithmic time
   c. Linear time
   d. Quadratic time
2. What is the worst-case time for binary search finding a single item in an array?
   a. Constant time
   b. Logarithmic time
   c. Linear time
   d. Quadratic time

3. Which of the following is not the required condition for binary search algorithm?
   a. The list must be sorted
   b. There should be the direct access to the middle element in any sub list
   c. There must be mechanism to delete and/or insert elements in list
   d. None of above
4. Which of the following is not a limitation of binary search algorithm?
   a. Must use a sorted array
   b. Requirement of sorted array is expensive when a lot of insertion and deletions are needed
   c. There must be a mechanism to access middle element directly
   d. Binary search algorithm is not efficient when the data elements are more than 1000.
5. Binary search algorithm cannot be applied to
   a. sorted linked list
   b. sorted binary trees
   c. sorted linear array
   d. pointer array
6. Which of the following case does not exist in complexity theory
   a. Best case
   b. Worst case
   c. Average case
   d. Null case
7. The Worst case occur in linear search algorithm when
   a. Item is somewhere in the middle of array
   b. Item is not in the array at all
   c. Item is the last element in the array
   d. Item is the last element in the array or is not there at all
8. The average case occur in linear search algorithm
   a. When item is somewhere in the middle of the array
   b. When item is not in the array at all
   c. When item is the last element in the array
   d. When item is the last element in the array or is not there at all
9. The complexity of the average case of an algorithm is
   a. Much more complicated to analyze then that of worst case
   b. Much more simpler to analyze than that of worst case
   c. Sometimes more complicated and some other times simpler than that of worst case
   d. None or above
10. The time complexity of linear search algorithm over an array of n element is
   a. O(log$_2$ n)   b. O(n)   c. O(n log$_2$ n)   d. O(n2)
11. The time required to search an element in a linked of length n is
   a. O(log$_2$ n)   b. O(n)   c. O(n log$_2$ n)   d. O(n2)
12. The complexity of linear search algorithm is
   a. O(n)
   b. O(log n)
   c. O(n2)
   d. O(n log n)
13. The complexity of binary search algorithm is
   a. O(n)
   b. O(log)
   c. O(n2)
   d. O(n log n)
14. A search begins the search with the element that is located in the middle of array
   a. Serial
   b. Random

c. Parallel
d. binary

15. The complexity of Bubble sort algorithm is
a. O(n)
b. O(log n)
c. $O(n^2)$
d. O(n log n)

16. The complexity of insertion sort algorithm is
a. O(n)
b. O(log n)
c. $O(n^2)$
d. O(n log n)

17. The complexity of merge sort algorithm is
a. O(n)
b. O(log n)
c. $O(n^2)$
d. $O(n.\log_2 n)$

18. Which of the following sorting algorithms does not have a worst case running time of $O(n^2)$
a. Insertion sort
b. Merge sort
c. Quick sort
d. Bubble sort

19. To sort many large objects or structures, it would be most efficient to place
a. Them in an array and sort the array
b. Pointers to them in an array and sort the array
c. Them in a linked list and sort the linked list
d. References to them in an array and sort the array.

20. Which of the following sorting procedure is the slowest?
a. Quick sort
b. Heap sort
c. Shell sort
d. Bubble sort

21. Which of the following sorting method is stable?
a. Straight insertion sort
b. Binary insertion sort
c. Shell sort
d. Heap sort

22. The number of swapping needed to sort the numbers 8, 22, 7, 9, 31, 19, 5, 13 in ascending order, using bubble sort is
a. 11
b. 12
c. 13
d. 14