Subject: - Data Structure using 'C' (22317)

SYLLABUS

Chapter No.	Name of chapter	Marks With Option
1	Introduction to data structure	12
2	Searching and sorting	22
3	Stack and queues	22
4	Linked list	24
5	Trees and Graphs	22
Total Marks: -		104

BOARD THEORY PAPER PATTERN FOR DSU (22317)

Q.1		Attempt any FIVE 5*2=10
	a)	Introduction to data structure
	b)	Stack and queues
	c)	Linked list
	d)	Stack and queues
	e)	Trees and Graphs
	f)	Searching and sorting
	g)	Stack and queues
Q.2		Attempt any THREE 3*4=12
	a)	Searching and sorting
	b)	Linked list
	c)	Stack and queues
	d)	Trees and Graphs

Q.3		Attempt any THREE 3*4=12
	a)	Introduction to data structure
	b)	Stack and queues
	c)	Searching and sorting
	d)	Trees and Graphs
	e)	Linked list
Q.4		Attempt any TWO 2*6=12
	a)	Searching and sorting
	b)	Trees and Graphs
	c)	Linked list
Q.5		Attempt any TWO 2*6=12
	a)	Stack and queues
	b)	Trees and Graphs
	c)	Linked list
Q.6		Attempt any TWO 2*6=12
	a)	Searching and sorting
	b)	Introduction to data structure
	c)	Linked list

CLASS TEST - I

PAPER PATTERN

COURSE: - Data Structure using 'C' (22317)

PROGRAMME: - Information Technology

Syllabus: -

Unit No.	Name of the Unit	Course Outcome (CO)
1	**Introduction to data structure**	**CO-317.1**
2	**Searching and sorting**	**CO-317.2**
3	**Stack and queues**	**CO-317.3**

Q.1	**Attempt any FOUR** **4*2=8Marks**	Course Outcome (CO)
a)	Introduction to data structure	CO-317.1
b)	Searching and sorting	CO-317.2
c)	Stack and queues	CO-317.3
d)	Introduction to data structure	CO-317.1
e)	Searching and sorting	CO-317.2
f)	Stack and queues	CO-317.3
Q.2	**Attempt any THREE** **3*4=12 Marks**	
a)	Introduction to data structure	CO-317.1
b)	Searching and sorting	CO-317.2

c)	Searching and sorting	CO-317.2
d)	Stack and queues	CO-317.3

CLASS TEST - II

PAPER PATTERN

COURSE: - Data Structure using 'C' (22317)

PROGRAMME: - Information Technology

Syllabus: -

Unit No.	Name of the Unit	Course Outcome (CO)
3	Stack and queues	CO-317.3
4	Linked list	CO-317.4
5	Trees and Graphs	CO-317.5

Q.1	Attempt any FOUR 4*2=8Marks	Course Outcome (CO)
a)	Stack and queues	(CO-317.3)
b)	Stack and queues	(CO-317.3)
c)	Linked list	(CO-317.4)
d)	Linked list	(CO-317.4)
e)	Trees and Graphs	(CO-317.5)
f)	Trees and Graphs	(CO-317.5)
Q.2	Attempt any THREE 3*4=12 Marks	
a)	Stack and queues	(CO-317.3)
b)	Linked list	(CO-317.4)
c)	Trees and Graphs	(CO-317.5)
d)	Trees and Graphs	(CO-317.5)

COURSE OUTCOME (CO)

PROGRAMME: - Information Technology

COURSE: - Data Structure using 'C' (22317)

CO. NO.	Course Outcome
CO-317.01	Perform basic operation on arrays.
CO-317.02	Apply different searching and sorting techniques.
CO-317.03	Implement basic operation on stack and queue using array representation.
CO-317.04	Implement basic operation on link list.
CO-317.05	Implement program to create and traverse tree to solve problems.

1. Introduction to data structure

Position in Question Paper **Total Marks-22**

Q.1. a) 2-Marks.
Q.1. b) 2-Marks.
Q.2. a) 4-Marks.
Q.3. d) 4-Marks.

Descriptive Question

1. Give classification of data structure.
2. Explain different approaches to design algorithm.
3. State different types of data types.
4. Compare linear and nonlinear data structure.
5. Explain time and space complexity in an algorithm.
6. Give four basic operation of data structure.
7. Define primitive and non-primitive data structure.
8. Describe abstract data type in detail
9. Describe Big ‘O’ notation.
10. Define data structure ? Why do you need data structure?

MCQ Question

(Total number of Question=Marks*3=6*3=18)

Note: Correct answer is marked with **bold.**

1. Which of these best describes an array?
   a) A data structure that shows a hierarchical behavior
   b) Container of objects of similar types
   c) Arrays are immutable once initialised
   d) Array is not a data structure
2. How do you initialize an array in C?
   a) int arr[3] = (1,2,3);
   c) int arr[3] = {1,2,3};
   b) int arr(3) = {1,2,3};
   d) int arr(3) = (1,2,3);
3. How do you instantiate an array in Java?
   a) int arr[] = new int(3);
   c) int arr[] = new int[3];
   b) int arr[];
   d) int arr() = new int(3);
4. Which of the following is the correct way to declare a multidimensional array in Java?
   a) int[] arr;
   c) int[][]arr;
   b) int arr[[]];
   d) int[[]] arr;
5. When does the ArrayIndexOutOfBoundsException occur?
   a) Compile-time
   c) Not an error
   b) Run-time
   d) Not an exception at all
6. Which of the following concepts make extensive use of arrays?
   a) Binary trees
   c) Caching
   b) Scheduling of processes
   d) Spatial locality
7. What are the advantages of arrays?
   a) Objects of mixed data types can be stored
   b) Elements in an array cannot be sorted
   c) Index of first element of an array is 1
   d) Easier to store elements of same data type

8. What are the disadvantages of arrays?
   a) Data structure like queue or stack cannot be implemented
   b) There are chances of wastage of memory space if elements inserted in an array are lesser than the allocated size
   c) Index value of an array can be negative
   d) Elements are sequentially accessed
9. Assuming int is of 4bytes, what is the size of int arr[15];?
   a) 15 c) 11
   b) 19 **d) 60**
10. In general, the index of the first element in an array is _____
   a) **0** c) 2
   b) -1 d) 1
11. Elements in an array are accessed _____
   a) randomly c) exponentially
   b) sequentially d) logarithmically
12. Recursion is a method in which the solution of a problem depends on _____
   a) Larger instances of different problems
   b) Larger instances of the same problem
   c) Smaller instances of the same problem
   d) Smaller instances of different problems
13. Which of the following problems can't be solved using recursion?
   a) Factorial of a number c) Length of a string
   b) Nth fibonacci number d) Problems without base case
14. Recursion is similar to which of the following?
   a) Switch Case c) If-else
   b) Loop d) if elif else
15. In recursion, the condition for which the function will stop calling itself is _____
   a) Best case **c) Base case**
   b) Worst case **d)** There is no such condition
16. Which of the following statements is true?
   a) Recursion is always better than iteration
   b) Recursion uses more memory compared to iteration

c) Recursion uses less memory compared to iteration

d) Iteration is always better and simpler than recursion

17. Suppose the first Fibonacci number is 0 and the second is 1. What is the sixth Fibonacci number?

a) 5
c) 7
b) 6
d) **8**

18. Which of the following is not a Fibonacci number?

a) 8
c) 55
b) 21
d) 14

2 Searching and sorting

Position in Question Paper **Total Marks-22**

Q.1. c) 2-Marks.
Q.2. b) 4-Marks.
Q.3. b) 4-Marks
Q.4. c) 6-Marks
Q.6. a) 6-Marks.

Descriptive Question

1. Define sorting. Enlist different methods.
2. Define internal and external sorting.
3. Explain efficiency of sorting algorithm.
4. Describe bubble sort with example.
5. Write an algorithm of bubble sort.
6. Describe the principle of selection sort with example.
7. Write an algorithm of selection sort.
8. Describe the principle of insertion sort with example.
9. Write an algorithm of insertion sort.
10. Describe the principle of radix sort with example.
11. Write an algorithm of radix sort.
12. Sort the following numbers using radix sort.
13. 10,5,99,105,55,100,135,141,137,200,199
14. Describe the principle of quick sort with example.
15. Write an algorithm of quick sort.
16. Give advantages and disadvantages of quick sort.
17. Give complexity of bubble sort .
18. Compare quick sort and radix sort.
19. Define searching. State two methods.
20. Differentiate between linear and binary search.
21. Explain linear search algorithm.
22. Describe linear search with example.

23. Explain binary search algorithm.
24. Describe binary search with example

MCQ Question

(Total number of Question=Marks*3=12*3=36)

Note: Correct answer is marked with bold

1. Where is linear searching used?
   a) When the list has only a few elements
   b) When performing a single search in an unordered list
   c) Used all the time
   d) When the list has only a few elements and When performing a single search in an unordered list
2. What is the best case for linear search?
   a) O(nlogn)
   b) O(logn)
   c) O(n)
   d) O(1)
3. What is the worst case for linear search?
   a) O(nlogn)
   b) O(logn)
   c) O(n)
   d) O(1)
4. What is the best case and worst case complexity of ordered linear search?
   a) O(nlogn), O(logn)
   b) O(logn), O(nlogn)
   c) O(n), O(1)
   d) O(1), O(n)
5. Which of the following is a disadvantage of linear search?
   a) Requires more space
   b) Greater time complexities compared to other searching algorithms
   c) Not easy to understand
   d) Not easy to implement
6. What is the advantage of recursive approach than an iterative approach?
   a) Consumes less memory
   b) Less code and easy to implement
   c) Consumes more memory
   d) More code has to be written
7. Given an input arr = {2,5,7,99,899}; key = 899; What is the level of recursion?
   a) 5
   b) 2
   c) 3
   d) 4
8. Given an array arr = {45,77,89,90,94,99,100} and key = 99; what are the mid values(corresponding array elements) in the first and second levels of recursion?

a) 90 and 99
c) 89 and 99
b) 90 and 94
d) 89 and 94

9. What is the worst case complexity of binary search using recursion?
a) O(nlogn)
c) O(n)
b) O(logn)
d) O(n2)

10. What is the average case time complexity of binary search using recursion?
a) O(nlogn)
c) O(n)
b) O(logn)
d) O(n2)

11. Which of the following is not an application of binary search?
a) To find the lower/upper bound in an ordered sequence
b) Union of intervals
c) Debugging
d) To search in unordered list

12. Binary Search can be categorized into which of the following?
a) Brute Force technique
c) Greedy algorithm
b) Divide and conquer
d) Dynamic programming

13. Given an array arr = {5,6,77,88,99} and key = 88; How many iterations are done until the element is found?
a) 1
c) 4
b) 3
d) **2**

14. Given an array arr = {45,77,89,90,94,99,100} and key = 100; What are the mid values(corresponding array elements) generated in the first and second iterations?
a) 90 and 99
c) 89 and 94
b) 90 and 100
d) 94 and 99

15. How many passes does an insertion sort algorithm consist of?
a) N
c) N+1
b) N-1
d) N2

16. Which of the following algorithm implementations is similar to that of an insertion sort?
a) Binary heap
c) Merge sort
b) Quick sort
d) Radix sort

17. What is the average case running time of an insertion sort algorithm?
a) O(N)
c) O(log N)
b) O(N log N)
d) O(N2)

18. Any algorithm that sorts by exchanging adjacent elements require O(N2) on average.
a) True b) False

19. What is the average number of inversions in an array of N distinct numbers?
a) N(N-1)/4 c) N(N-1)/2
b) N(N+1)/2 d) N(N-1)/3

20. What is the average number of inversions in an array of N distinct numbers?
a) N(N-1)/4 **c) N(N-1)/2**
b) N(N+1)/2 d) N(N-1)/3

21. What will be the number of passes to sort the elements using insertion sort?
14, 12,16, 6, 3, 10
a) 6 c) 7
b) **5** d) 1

22. For the following question, how will the array elements look like after second pass?
34, 8, 64, 51, 32, 21
a) 8, 21, 32, 34, 51, 64 c) 8, 34, 51, 64, 32, 21
b) 8, 32, 34, 51, 64, 21 **d) 8, 34, 64, 51, 32, 21**

23. What is an in-place sorting algorithm?
a) It needs O(1) or O(logn) memory to create auxiliary locations
b) The input is already sorted and in-place
c) It requires additional storage
d) It requires additional space

24. What is the worst case complexity of selection sort?
a) O(nlogn) c) O(n)
b) O(logn) **d) O(n2)**

25. What is the advantage of selection sort over other sorting techniques?
a) It requires no additional storage space
b) It is scalable
c) It works best for inputs which are already sorted
d) It is faster than any other sorting technique

26. What is the average case complexity of selection sort?
a) O(nlogn) c) O(n)
b) O(logn) **d) O(n2)**

27. Merge sort uses which of the following technique to implement sorting?

a) backtracking
c) divide and conquer
b) greedy algorithm
d) dynamic programming

28. What is the auxiliary space complexity of merge sort?
a) O(1)
c) O(n)
b) O(log n)
d) O(n log n)

29. What is the worst case time complexity of merge sort?
a) O(n log n)
c) O(n2 log n)
b) O(n2)
d) O(n log n2)

30. Which of the following sorting algorithms is the fastest?
a) Merge sort
c) Insertion sort
b) Quick sort
d) Shell sort

31. Quick sort follows Divide-and-Conquer strategy.
a) True
b) False

32. Which of the following methods is the most effective for picking the pivot element?
a) first element
c) median-of-three partitioning
b) last element
d) random element

33. Find the pivot element from the given input using median-of-three partitioning method.
8, 1, 4, 9, 6, 3, 5, 2, 7, 0.
a) 8
c) 9
b) 7
d) **6**

34. Which is the safest method to choose a pivot element?
a) choosing a random element as pivot
b) choosing the first element as pivot
c) choosing the last element as pivot
d) median-of-three partitioning method

35. What is the average running time of a quick sort algorithm?
a) O(N2)
c) O(N log N)
b) O(N)
d) O(log N)

36. Which of the following sorting algorithms is used along with quick sort to sort the sub arrays?
a) Merge sort
c) Insertion sort
b) Shell sort
d) Bubble sort

37.Quick sort uses join operation rather than merge operation.

a) true b) false

38. How many sub arrays does the quick sort algorithm divide the entire array into?

a) one **b) two**

c) three d) four

3. Stack and queues

Position in Question Paper **Total Marks-22**

Q.1. c) 2-Marks.
Q.2. b) 4-Marks.
Q.3. b) 4-Marks
Q.4. c) 6-Marks
Q.6. a) 6-Marks.

Descriptive Questions-

1. State the principle of stack with basic operation.
2. Define stack.
3. State importance of top pointer in stack.
4. Explain stack as an abstract data type.
5. Explain the condition stack overflow and underflow.
6. Explain push and pop operation in stack
7. Explain application of stack.
8. Evaluate the following postfix expressions.
9. 5,4,6,+,*,4,9,3,/,+,*
10. Convert the following infix expression into postfix expression. (A+B*C/D-E+F/G/(H+I))
11. Write an algorithm to convert infix to postfix expression.
12. Convert the following expression into prefix. (A-B/C)*(D*E-F)
13. Explain the concept of recursion.
14. How the problem of tower of honoi is solved?
15. Define queue. Explain the term front and rear.
16. Compare stack and queue.
17. Write a program for insertion and deletion of queue.
18. Explain queue implementation using linked list.
19. Explain queue as an abstract data type.
20. Draw and explain circular queue in detail.
21. Explain the insertion and deletion of circular queue.

MCQ Question

(Total number of Question=Marks*3=20*3=60)

Note: Correct answer is marked with bold

1. Process of inserting an element in stack is called_____
   a) Create
   b) Push
   c) Evaluation
   d) Pop
2. Process of removing an element from stack is called _____
   a) Create
   b) Push
   c) Evaluation
   d) Pop
3. In a stack, if a user tries to remove an element from an empty stack it is called _____
   a) Underflow
   b) Empty collection
   c) Overflow
   d) Garbage Collection
4. Pushing an element into stack already having five elements and stack size of 5, then stack becomes _____
   a) Overflow
   b) Crash
   c) Underflow
   d) User flow
5. Entries in a stack are "ordered". What is the meaning of this statement?
   a) A collection of stacks is sortable
   b) Stack entries may be compared with the '<' operation
   c) The entries are stored in a linked list
   d) There is a Sequential entry that is one by one
6. Which of the following is not the application of stack?
   a) A parentheses balancing program
   b) Tracking of local variables at run time
   c) Compiler Syntax Analyzer
   d) Data Transfer between two asynchronous process
7. 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 ONE TIME when the algorithm analyzes: (()(())(())))?
   a) 1
   b) 2
   c) 3
   d) 4 or more

8. What is the value of the postfix expression 6 3 2 4 + – *?
   a) 1 **c)** 74
   b) 40 **d) -18**
9. Here is an infix expression: 4 + 3*(6*3-12). Suppose that we are using the usual stack algorithm to convert the expression from infix to postfix notation. The maximum number of symbols that will appear on the stack AT ONE TIME during the conversion of this expression?
   a) 1 **c)** 3
   b) 2 **d) 4**
10. What data structure is used when converting an infix notation to prefix notation?
   a) Stack **c)** B-Trees
   b) Queue **d)** Linked-list
11. Out of the following operators (^, *, +, &, $), the one having highest priority is _____
   a) + **c)** ^
   b) $ d) &
12. How many stacks are required for evaluation of prefix expression?
   a) one **c)** three
   b) two **d)** four
13. While evaluating a prefix expression, the string is read from?
   a) left to right **c)** center to right
   b) right to left **d)** center to left to right
14. How many types of input characters are accepted by this algorithm?
   a) one **c) three**
   b) two **d)** four
15. What determines the order of evaluation of a prefix expression?
   a) precedence and associativity **c)** associativity only
   b) precedence only **d)** depends on the parser
16. Find the output of the following prefix expression
   *+2-2 1/-4 2+-5 3 1
   a) 2 c) 10
   b) 12 d) 4

17. Using the evaluation of prefix algorithm, evaluate +-9 2 7.
a) 10 **c)** 17
b) 4 **d) 14**

18. If -*+abcd = 11, find a, b, c, d using evaluation of prefix algorithm.
a) a=2, b=3, c=5, d=4 c) a=5, b=4, c=7,d=5
b) a=1, b=2, c=5, d=4 d) a=1, b=2, c=3, d=4

19. The optimal data structure used to solve Tower of Hanoi is _____
a) Tree **c)** Priority queue
b) Heap **d)** Stack

20. Which among the following is not a palindrome?
a) Madam c) Malayalam
b) Dad **d) Maadam**

21. What is the number of moves required to solve Tower of Hanoi problem for k disks?
a) 2k – 1 **c)** $2^k + 1$
b) 2k + 1 **d)** $\mathbf{2^k - 1}$

22. What is the other name for a postfix expression?
a) Normal polish Notation c) Warsaw notation
b) Reverse polish Notation d) Infix notation

23. Which of the following is an example for a postfix expression?
a) a*b(c+d) c) +ab
b) abc*+de-+ d) a+b-c

24. What is the time complexity of evaluation of postfix expression algorithm?
a) O (N) **c)** O (N^2)
b) O (N log N) **d)** O (M log N)

25. In Postfix expressions, the operators come after the operands.
a) True
b) False

27. Which of these operators have the highest order of precedence?
a) '(' and ')' **c) '~' and '^'**
b) '*' and '/' d) '+' and '-'

28. Which of the following is not an application of stack?
a) evaluation of postfix expression
b) conversion of infix to postfix expression
c) balancing symbols
d) line at ticket counter

29. while evaluating a postfix expression, when an operator is encountered, what is the correct operation to be performed?
    a) push it directly on to the stack
    b) pop 2 operands, evaluate them and push the result on to the stack
    c) pop the entire stack
    d) ignore the operator
30. Which of the following statement is incorrect?
    a) Postfix operators use value to their right
    b) Postfix operators use value to their left
    c) Prefix operators use value to their right
    d) In postfix expression, operands are followed by operators
31. What is the result of the given postfix expression? abc*+ where a=1, b=2, c=3.
    a) 4 c) 6
    b) 5 d) **7**
32. What is the result of the following postfix expression?
    ab*cd*+ where a=2,b=2,c=3,d=4.
    a) 16 c) 14
    b) 12 d) 10
33. Evaluate the postfix expression ab + cd/- where a=5, b=4, c=9, d=3.
    a) 23 **c) 6**
    b) 15 d) 10
34. Reversing a word using stack can be used to find if the given word is a palindrome or not.
    a) **True** b) false
35. Which is the most appropriate data structure for reversing a word?
    a) queue c) tree
    b) stack d) graph
36. Operations required for reversing a word or a string using stack are push() and pop().
    a) True b) False
37. What is the time complexity of reversing a word using stack algorithm?
    a) O (N log N) **c) O (N)**
    b) O (N^2) d) O (M log N)
38. What will be the word obtained if the word "abbcabb" is reversed using a stack?

a) bbabbca
c) bbacbba
b) abbcabb
d) bbacabb

39. A linear list of elements in which deletion can be done from one end (front) and insertion can take place only at the other end (rear) is known as _____
a) Queue
c) Tree
b) Stack
d) Linked list

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

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

42. Circular Queue is also known as _____
a) Ring Buffer
c) Rectangle Buffer
b) Square Buffer
d) Curve Buffer

43. If the elements "A", "B", "C" and "D" are placed in a queue and are deleted one at a time, in what order will they be removed?
a) ABCD
c) DCAB
b) DCBA
d) ABDC

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

45. 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

46. Queues serve major role in _____
a) Simulation of recursion
b) Simulation of arbitrary linked list
c) Simulation of limited resource allocation
d) Simulation of heap sort

47. Which of the following is not the type of queue?

a) Ordinary queue
c) Circular queue
b) Single ended queue
d) Priority queue

48. Express -15 as a 6-bit signed binary number.
a) 001111
c) 101110
b) 101111
d) 001110

49. Which is the predefined method available in Java to convert decimal to binary numbers?
a) toBinaryInteger(int)
c) toBinaryNumber(int)
b) toBinaryValue(int)
d) toBinaryString(int)

50. What is the time complexity for converting decimal to binary numbers?
a) O(1)
c) O(logn)
b) O(n)
d) O(nlogn)

51. Which of the following data structure is used to convert postfix expression to infix expression?
a) Stack
c) Linked List
b) Queue
d) Heap

52. The postfix expression abc+de/*- is equivalent to which of the following infix expression?
a) abc+-de*/
c) a-(b+c)*(d/e)
b) (a+b)-d/e*c
d) abc+*-(d/e)

53. The equivalent infix expression and value for the postfix form 1 2 + 3 * 4 5 * – will be _____
a) 1 + 2 * 3 – 4 * 5 and -13
b) (2 + 1) * (3 – 4) * 5 and 13
c) 1 + 2 * (3 – 4) * 5 and -11
d) (1 + 2) * 3 – (4 * 5) and -11

54. What is the value of the postfix expression 2 3 + 4 5 6 – – *
a) 19
c) -4
b) 21
d) 25

55. The prefix expression of the postfix expression AB+CD-* is _____
a) (A+B)*(C-D)
c) A+*BCD-
b) +AB*-CD
d) *+AB-CD

56. Consider the postfix expression 4 5 6 a b 7 8 a c, where a, b, c are operators. Operator a has higher precedence over operators b and c. Operators b and c are right associative. Then, equivalent infix expression is
    a) 4 a 5 6 b 7 8 a c
    c) 4 b 5 a 6 c 7 a 8
    b) 4 a 5 c 6 b 7 a 8
    d) 4 a 5 b 6 c 7 a 8
57. To convert the postfix expression into the infix expression we use stack and scan the postfix expression from left to right.
    a) True
    b) False
58. The result of the postfix expression 5 3 * 9 + 6 / 8 4 / + is _____
    a) 8
    c) 10
    b) **6**
    d) 9
59. How many stacks are required for reversing a word algorithm?
    a) one
    c) three
    b) two
    d) four
60. Evaluate and write the result for the following postfix expression abc*+de*f+g*+ where a=1, b=2, c=3, d=4, e=5, f=6, g=2.
    a) 61
    c) 60
    b) 59
    d) 55

4. Linked list

Position in Question Paper **Total Marks-24**

Q.1. d) 2-Marks.
Q.1. e) 2-Marks.
Q.2. b) 4-Marks.
Q.3. b) 4-Marks
Q.5. a) 6-Marks.
Q.6. a) 6-Marks.

Descriptive Questions-

1. List types of linked list and state the operation performed on linked list.
2. Write an algorithm to insert new node at the beginning, middle and end of linked list.
3. Define node, null pointer, empty list, data, nextpointer, address.
4. Explain the operation on searching a desired node in linked list.
5. Explain the linked list as an abstract data type.
6. Write program to delete node in linked list.
7. Describe the structure of circular linked list.
8. Define dynamic memory allocation. Give its importance.
9. Write an algorithm to count number of nodes in singly link list.
10. Draw representation of singly linked list.
11. Draw representation of doubly linked list.
12. Draw representation of circular linked list.
13. With example describe how circular linked list works when a node is deleted from beginning of list.
14. Compare linear linked list, circular linked list.

MCQ Question

(Total number of Question=Marks*3=16*3=48)

Note: Correct answer is marked with bold

1. Which of the following is not a disadvantage to the usage of array?
   a) Fixed size
   b) There are chances of wastage of memory space if elements inserted in an array are lesser than the allocated size
   c) Insertion based on position
   d) Accessing elements at specified positions
2. What is the time complexity of inserting at the end in dynamic arrays?
   a) O(1)
   b) O(n)
   c) O(logn)
   d) Either O(1) or O(n)
3. What is the time complexity to count the number of elements in the linked list?
   a) O(1)
   b) O(n)
   c) O(logn)
   d) $O(n^2)$
4. What is the space complexity for deleting a linked list?
   a) O(1)
   b) O(n)
   c) Either O(1) or O(n)
   d) O(logn)
5. Which of these is not an application of a linked list?
   a) To implement file systems
   b) For separate chaining in hash-tables
   c) To implement non-binary trees
   d) Random Access of elements
6. Which of the following is false about a doubly linked list?
   a) We can navigate in both the directions
   b) It requires more space than a singly linked list
   c) The insertion and deletion of a node take a bit longer
   d) Implementing a doubly linked list is easier than singly linked list
7. What is a memory efficient double linked list?
   a) Each node has only one pointer to traverse the list back and forth
   b) The list has breakpoints for faster traversal
   c) An auxiliary singly linked list acts as a helper list to traverse through the doubly

linked list

d) A doubly linked list that uses bitwise AND operator for storing addresses

8. How do you calculate the pointer difference in a memory efficient double linked list?
   a) head xor tail
   b) pointer to previous node xor pointer to next node
   c) pointer to previous node – pointer to next node
   d) pointer to next node – pointer to previous node
9. What is the worst case time complexity of inserting a node in a doubly linked list?
   a) O(nlogn) **c)** O(n)
   b) O(logn) **d)** O(1)
10. A linear collection of data elements where the linear node is given by means of pointer is called?
   a) Linked list **c)** Primitive list
   b) Node list **d)** Unordered list
11. In linked list each node contains a minimum of two fields. One field is data field to store the data second field is?
   a) Pointer to character **c) Pointer to node**
   b) Pointer to integer d) Node
12. What would be the asymptotic time complexity to add a node at the end ofsingly linked list, if the pointer is initially pointing to the head of the list?
   a) O(1) **c) θ(n)**
   b) O(n) d) θ(1)
13. What would be the asymptotic time complexity to insert an element at the front of the linked list (head is known)?
   a) O(1) c) $O(n^2)$
   b) O(n) d) $O(n^3)$
14. What would be the asymptotic time complexity to find an element in the linked list?
   a) O(1) c) $O(n^2)$
   b) O(n) d) $O(n^4)$
15. What would be the asymptotic time complexity to find an element in the linked list?
   a) O(1) c) $O(n^2)$
   b) O(n) d) $O(n^4)$
16. The concatenation of two lists can be performed in O(1) time. Which of the following variation of the linked list can be used?
   a) Singly linked list

b) Doubly linked list

c) Circular doubly linked list

d) Array implementation of list

17. Which of the following is false about a doubly linked list?

a) We can navigate in both the directions

b) It requires more space than a singly linked list

c) The insertion and deletion of a node take a bit longer

d) Implementing a doubly linked list is easier than singly linked list

18. What is a memory efficient double linked list?

a) Each node has only one pointer to traverse the list back and forth

b) The list has breakpoints for faster traversal

c) An auxiliary singly linked list acts as a helper list to traverse through the doubly linked list

d) A doubly linked list that uses bitwise AND operator for storing addresses

19. How do you calculate the pointer difference in a memory efficient double linked list?

a) head xor tail

b) pointer to previous node xor pointer to next node

c) pointer to previous node – pointer to next node

d) pointer to next node – pointer to previous node

20. What is the worst case time complexity of inserting a node in a doubly linked list?

a) O(nlogn) **c) O(n)**

b) O(logn) d) O(1)

21. What differentiates a circular linked list from a normal linked list?

a) You cannot have the 'next' pointer point to null in a circular linked list

b) It is faster to traverse the circular linked list

c) You may or may not have the 'next' pointer point to null in a circular linked list

d) Head node is known in circular linked list

22. What is the time complexity of searching for an element in a circular linked list?

a) O(n) c) O(1)

b) O(nlogn) d) $O(n^2)$

23. Which of the following application makes use of a circular linked list?

a) Undo operation in a text editor

b) Recursive function calls

c) Allocating CPU to resources

d) Implement Hash Tables

24. 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) We can traverse the whole circular linked list by starting from any point

25. 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) Circular linked list itself represents a cycle. So no new cycles cannot be generated

26. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is?

a) log 2 n

b) $^{n}/_{2}$

c) log 2 n – 1

d) **n**

27. Given pointer to a node X in a singly linked list. Only one pointer is given, pointer to head node is not given, can we delete the node X from given linked list?

a) Possible if X is not last node

b) Possible if size of linked list is even

c) Possible if size of linked list is odd

d) Possible if X is not first node

28. You are given pointers to first and last nodes of a singly linked list, which of the following operations are dependent on the length of the linked list?

a) Delete the first element

b) Insert a new element as a first element

c) Delete the last element of the list

d) Add a new element at the end of the list

29. What is a skip list?

a) a linkedlist with size value in nodes

b) a linkedlist that allows faster search within an ordered sequence

c) a linkedlist that allows slower search within an ordered sequence
d) a tree which is in the form of linked list

30. Skip lists are similar to which of the following datastructure?
a) stack
b) heap
c) binary search tree
d) balanced binary search tree

31. What is the time complexity improvement of skip lists from linked lists in insertion and deletion?
a) O(n) to O(logn) where n is number of elements
b) O(n) to O(1) where n is number of elements
c) no change
d) O(n) to $O(n^2)$ where n is number of elements

32. To which datastructure are skip lists similar to in terms of time complexities in worst and best cases?
a) balanced binary search trees
b) binary search trees
c) binary trees
d) linked lists

33. The nodes in a skip list may have many forward references. their number is determined
a) probabilistically
b) randomly
c) sequentially
d) orthogonally

34. Are the below statements true about skiplists?
In a sorted set of elements skip lists can implement the below operations
i. given a element find closest element to the given value in the sorted set in O(logn)
ii. find the number of elements in the set whose values fall a given range in O(logn)
a) **true**
b) false

35. How to maintain multi-level skip list properties when insertions and deletions are done?
a) design each level of a multi-level skip list with varied probabilities
b) that cannot be maintained
c) rebalancing of lists
d) reconstruction

36. Is a skip list like balanced tree?
a) true
b) false

37. What is indexed skip list?
    a) it stores width of link in place of element
    b) it stores index values
    c) array based linked list
    d) indexed tree
38. What kind of linked list is best to answer questions like "What is the item at position n?"
    a) Singly linked list
    b) Doubly linked list
    c) Circular linked list
    d) Array implementation of linked list
39. Linked lists are not suitable for the implementation of _____
    a) Insertion sort
    b) Radix sort
    c) Polynomial manipulation
    d) Binary search
40. Linked list is considered as an example of _____ type of memory allocation.
    a) Dynamic
    b) Static
    c) Compile time
    d) Heap
41. In Linked List implementation, a node carries information regarding _____
    a) Data
    b) Link
    c) Data and Link
    d) Node
42. Linked list data structure offers considerable saving in _____
    a) Computational Time
    b) Space Utilization
    c) Space Utilization and Computational Time
    d) Speed Utilization
43. Which of the following points is/are not true about Linked List data structure when it is compared with an array?
    a) Arrays have better cache locality that can make them better in terms of performance
    b) It is easy to insert and delete elements in Linked List
    c) Random access is not allowed in a typical implementation of Linked Lists
    d) Access of elements in linked list takes less time than compared to arrays
44. Which of the following sorting algorithms can be used to sort a random linked list with minimum time complexity?

a) Insertion Sort
b) Quick Sort
c) Heap Sort
d) **Merge Sort**

45. What is a hash table?
a) A structure that maps values to keys
b) A structure that maps keys to values
c) A structure used for storage
d) A structure used to implement stack and queue

46. If several elements are competing for the same bucket in the hash table, what is it called?
a) Diffusion
b) Replication
c) Collision
d) Duplication

47. What is direct addressing?
a) **Distinct array position for every possible key**
b) Fewer array positions than keys
c) Fewer keys than array positions
d) Same array position for all keys

48. What is the search complexity in direct addressing?
a) O(n)
b) O(logn)
c) O(nlogn)
d) O(1)

5. Tree and Graph

Position in Question Paper **Total Marks-22**

Q.1. f) 2-Marks.
Q.2. d) 4-Marks.
Q.3. d) 4-Marks
Q.5. d) 6-Marks.
Q.6. d) 6-Marks.

Descriptive Questions-

1. Define the term tree
2. Describe in brief the terms related to binary tree:
   root, parent, child, siblings, path, degree of node ,leaf node ,level, depth,
3. degree of tree, height of tree, ancestor/descendent node
4. Explain height of tree.
5. Explain the binary tree with suitable example and diagram.
6. Describe with an example sequential representation of tree in memory.
7. Describe with an example linked representation of trees in memory.
8. Define the term general tree.
9. Compare general tree and binary tree.
10. Write a c program to count nodes in binarytree.
11. Enlist the type of binary tree and give there meaning.
12. Define the tree traversal.list different types.
13. Write an algorithm to for inorder traversal.
14. Write an algorithm to for preorder traversal.
15. Write an algorithm to for postorder traversal.
16. Enlist operation on trees.
17. Construct binary tree for the following data
    10,3,15,22,6,45,65,23,78,34,5
18. Describe expression tree with example.
19. Define following terms:

- graph,
- undirected graph,
- directed graph,
- complete graph,
- weighted graph,
- path,
- sink,
- articulation point,
- cycle,
- subgraph,
- connected graph ,
- componenet,
- degree of vertex,
- multigraph,
- self loop,
- spanning tree,
- successor,
- predecessor.

MCQ Question

(Total number of Question=Marks*3=16*3=48)

1. The number of edges from the root to the node is called _____ of the tree.
   a) Height
   c) Length
   b) Depth
   d) Width

2. The number of edges from the node to the deepest leaf is called _____ of the tree.
   a) Height
   c) Length
   b) Depth
   d) Width

3. What is a full binary tree?
   a) **Each node has exactly zero or two**
   b) Each node has exactly two children
   c) All the leaves are at the same level
   d) Each node has exactly one or two children

4. What is a complete binary tree?
   a) Each node has exactly zero or two children
   b) A binary tree, which is completely filled, with the possible exception of the bottom level, which is filled from right to left
   c) A binary tree, which is completely filled, with the possible exception of the bottom level, which is filled from left to right
   d) A tree In which all nodes have degree 2

5. What is the average case time complexity for finding the height of the binary tree?
   a) h = O(loglogn)
   c)h = O(n)
   b)h = O(nlogn)
   d)h = O(log n)

6. Which of the following is not an advantage of trees?
   a)Hierarchical structure
   b)Faster search
   c)Router algorithms
   d)Undo/Redo operations in a notepad

7. In a full binary tree if number of internal nodes is I, then number of leaves L are?

a)L = 2*I
b)L = I + 1
c)L = I – 1
d) L = 2*I – 1

8. In a full binary tree if number of internal nodes is I, then number of nodes N are?
a)N = 2*I
b)N = I + 1
c)N = I – 1
d) N = 2*I + 1

9. In a full binary tree if there are L leaves, then total number of nodes N are?
a)N = 2*L
b)N = L + 1
c)N = L – 1
d) N = 2*L – 1

10. Which of the following is incorrect with respect to binary trees?
a) Let T be a binary tree. For every $k \geq 0$, there are no more than 2k nodes in level k
b) Let T be a binary tree with λ levels. Then T has no more than $2^{\lambda - 1}$ nodes
c) Let T be a binary tree with N nodes. Then the number of levels is at least ceil(log (N + 1))
d) Let T be a binary tree with N nodes. Then the number of levels is at least floor(log (N + 1))

11. Which of the following is false about a binary search tree?
a) The left child is always lesser than its parent
b) The right child is always greater than its parent
c) The left and right sub-trees should also be binary search trees
d) In order sequence gives decreasing order of elements

12. What is the speciality about the inorder traversal of a binary search tree?
a) It traverses in a non increasing order
b) It traverses in an increasing order
c) It traverses in a random fashion
d) It traverses based on priority of the

13. What does the following piece of code

```
public void func(Tree root)
{
        func(root.left());
        func(root.right());
        System.out.println(root.d
        ata());
```

a)Preordertraversal
b)Inordertraversal
c)Postordertraversal
d) Level order traversal

14. What does the following piece of code do?

```
public void func(Tree root)
{
        System.out.println(root.data());

        func(root.left());
        func(root.right());
}
```

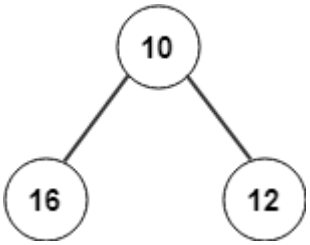
a)Preordertraversal
b)Inordertraversal
c)Postordertraversal
d)Level order traversal

15. What are the worst case and average case complexities of a binary search tree?
a)O(n),O(n)
b)O(logn),O(logn)
c)O(logn),O(n)
d)O(n), O(logn)

16. What are the conditions for an optimal binary search tree and what is its advantage?
a)The tree should not be modified and you should know how often the keys are accessed ,it improves the lookup cost
b)You should know the frequency of access of the keys, improves the lookup time
c)The tree can be modified and you should know the number of elements in the tree beforehand, it improves the deletion time
d)The tree should be just modified and improves the lookup time

17. What is the maximum number of children that a binary tree node can have?
a)0
b)1
c) 2
d) 3

18. The following given tree is an example for?



a)Binary tree
b)Binary search tree
c) Fibonacci tree
d)AVL tree

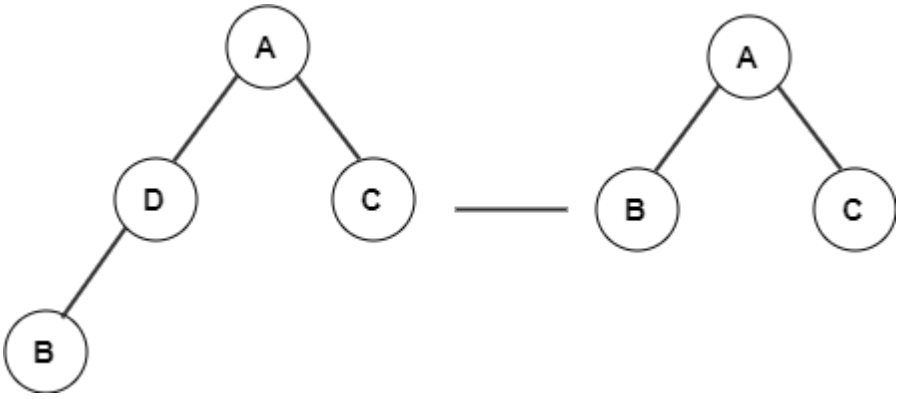
19. A binary tree is a rooted tree but not an ordered tree.
    a) true **b) false**

20. How many common operations are performed in a binary tree?
    a) 1 **c) 3**
    b) 2 d) 4

21. What is the traversal strategy used in the binary tree?
    a) depth-first traversal c) random traversal
    b) breadth-first traversal d) Priority traversal

22. How many types of insertion are performed in a binary tree?
    a) 1 c) 3
    b) 2 d) 4

23. What operation does the following diagram depict?



    a) inserting a leaf node
    b) inserting an internal node
    c) deleting a node with 0 or 1 child
    d) deleting a node with 2 children

24. General ordered tree can be encoded into binary trees.
    a) **true** b) false

25. How many bits would a succinct binary tree occupy?
    a) n+O(n) c) n/2
    b) 2n+O(n) d) n

26. How many orders of traversal are applicable to a binary tree (In General)?

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

27. If binary trees are represented in arrays, what formula can be used to locate a left child, if the node has an index i?

a)2i+1
b)2i+2
c) 2i
d)4i

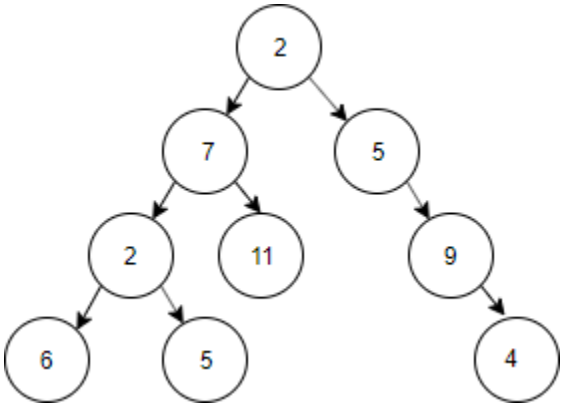
28. Using what formula can a parent node be located in an array?

a) (i+1)/2
b) (i-1)/2
c) i/2
d) 2i/2

29. Which of the following properties are obeyed by all three tree – traversals?

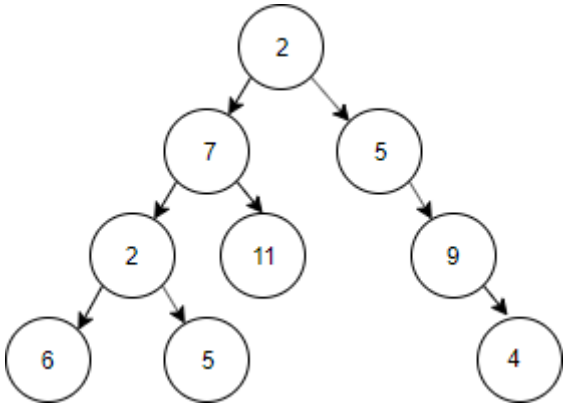
a) Left subtrees are visited before right subtrees
b) Right subtrees are visited before left subtrees
c) Root node is visited before left subtree
d) Root node is visited before right subtree

30. For the tree below, write the in-order traversal.



a) 6, 2, 5, 7, 11, 2, 5, 9, 4
b) 6, 5, 2, 11, 7, 4, 9, 5, 2
c) 2, 7, 2, 6, 5, 11, 5, 9, 4
d) 2, 7, 6, 5, 11, 2, 9, 5, 4

31. For the tree below, write the pre-order traversal.



a) 2, 7, 2, 6, 5, 11, 5, 9, 4
b) 2, 7, 5, 2, 6, 9, 5, 11, 4
c) 2, 5, 11, 6, 7, 4, 9, 5, 2
d) 2, 7, 5, 6, 11, 2, 5, 4, 9

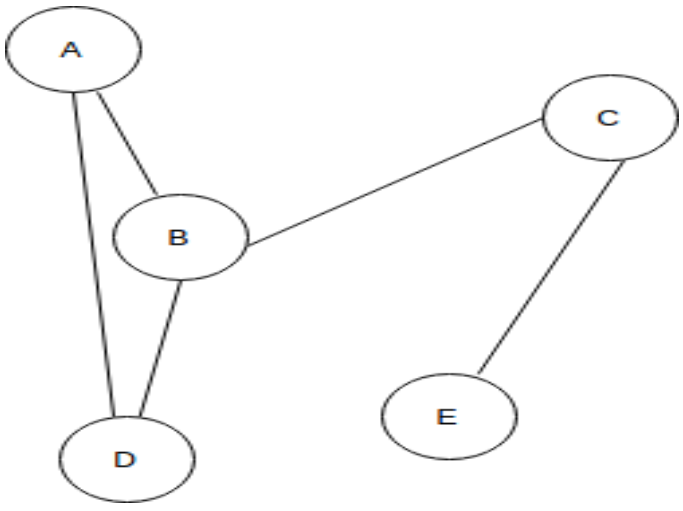
32. The post-order traversal of a binary tree is O P Q R S T. Then possible pre-order traversal will be _____

a) T Q R S O P
b) T O Q R P S
c) T Q O P S R
d) T Q O S P R

33. Which of the following statements for a simple graph is correct?

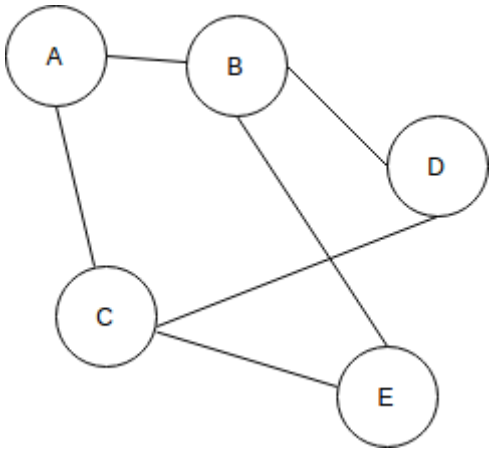
a) **Every path is a trail**
b) Every trail is a path
c) Every trail is a path as well as every path is a trail
d) Path and trail have no relation

34. In the given graph identify the cut vertices.



a) B and E
b) C and D
c) A and E
d) C and B

35. For the given graph(G), which of the following statements is true?



a) G is a complete graph
b) G is not a connected graph
c) The vertex connectivity of the graph is 2
d) The edge connectivity of the graph is 1

36. What is the number of edges present in a complete graph having n vertices?
a) (n*(n+1))/2
b) (n*(n-1))/2
c) n
d) Information given is insufficient

37. A connected planar graph having 6 vertices, 7 edges contains _____regions.
a)15
b)3
c)1
d)11

38. Which of the following properties does a simple graph not hold?
a) Must be connected
b) Must be unweighted
c) Must have no loops or multiple edges
d) Must have no multiple edges

39. What is the maximum number of edges in a bipartite graph having 10 vertices?
a) 24
b) 21
c) 25
d) 16

40. What would be the number of zeros in the adjacency matrix of the given graph?

a) 10
b) **6**
c) 16
d) 0

41. The time complexity to calculate the number of edges in a graph whose information in stored in form of an adjacency matrix is _____

a) O(V)
b) $O(E^2)$
c) O(E)
d) $O(V^2)$

42. For the adjacency matrix of a directed graph the row sum is the_____degree and the column sum is the _____degree.

a) in, out
b) out, in
c) in, total
d) total, out

43. 7. On which of the following statements does the time complexity of checking if an edge exists between two particular vertices is not, depends?

a) Depends on the number of edges
b) Depends on the number of vertices
c) Is independent of both the number of edges and vertices
d) It depends on both the number of edges and vertices

44. Given an adjacency matrix A = [[0, 1, 1], [1, 0, 1], [1, 1, 0]], The total no. of ways in which every vertex can walk to itself using 2 edges is_____

a) 2
b) 4
c) 6
d) 8

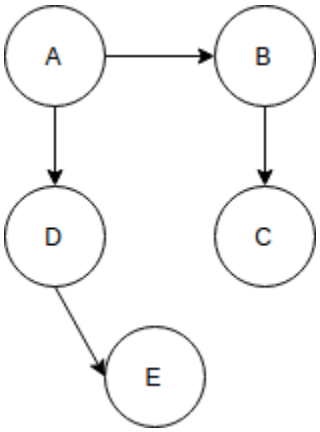
46.A graph having an edge from each vertex to every other vertex is called a _____

a) Tightly Connected
b) Strongly Connected
c) Weakly Connected
d) Loosely Connected

48. What is the number of unlabeled simple directed graph that can be made with 1 or 2 vertices?

a) 2
b) **4**
c) 5
d) 9

47. What would be the DFS traversal of the given Graph?



a) ABCED
b) AEDCB
c) EDCBA
d) ADECB