DATA STRUCTURES:

LINKED LISTS:

1) **What is the worst case time complexity of inserting a node in a doubly linked list?**

a) O(nlogn)

b) O(logn)

c) O(n)

d) O(1)

Answer: c

Explanation: In the worst case, the position to be inserted maybe at the end of the list, hence you have to traverse through the entire list to get to the correct position, hence O(n).

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

a) O(n)

b) O(nlogn)

c) O(1)

d) O(n2)

Answer: a

Explanation: In the worst case, you have to traverse through the entire list of n elements.

3) **A linear collection of data elements where the linear node is given by means of pointer is called?**

a) Linked list

b) Node list

c) Primitive list

d) Unordered list

Answer: a

Explanation: In Linked list each node has its own data and the address of next node. These nodes are linked by using pointers. Node list is an object that consists of a list of all nodes in a document with in a particular selected set of nodes.

4)Consider an implementation of unsorted singly linked list. Suppose it has its representation with a head pointer only.

Given the representation, which of the following operation can be implemented in O(1) time?

i) Insertion at the front of the linked list

ii) Insertion at the end of the linked list

iii) Deletion of the front node of the linked list

iv) Deletion of the last node of the linked list

a) I and II

b) I and III

c) I, II and III

d) I, II and IV

Answer: b

Explanation: We know the head node in the given linked list. Insertion and deletion of elements at the front of the linked list completes in O (1) time whereas for insertion and deletion at the last node requires to traverse through every node in the linked list. Suppose there are n elements in a linked list, we need to traverse through each node. Hence time complexity becomes O(n).

5) In linked list each node contain minimum of two fields. One field is data field to store the data second field is?

a) Pointer to character

b) Pointer to integer

c) Pointer to node

d) Node

Answer: c

Explanation: Each node in a linked list contains data and a pointer (reference) to the next node. Second field contains pointer to node.

6) What would be the asymptotic time complexity to add a node at the end of singly linked list, if the pointer is initially pointing to the head of the list?

a) O(1)

b) O(n)

c) θ(n)

d) θ(1)

Answer: c

Explanation: In case of a linked list having n elements, we need to travel through every node of the list to add the element at the end of the list. Thus asymptotic time complexity is θ(n).

7) What would be the asymptotic time complexity to insert an element at the front of the linked list (head is known)?

a) O(1)

b) O(n)

c) O(n_2)

d) O(n3)

Answer: b

Explanation: To add an element at the front of the linked list, we will create a new node which holds the data to be added to the linked list and pointer which points to head position in the linked list. The entire thing happens within O (1) time. Thus the asymptotic time complexity is O (1).

8)What would be the asymptotic time complexity to insert an element at the second position in the linked list?

a) O(1)

b) O(n)

c) O(n2)

d) O(n3)

Answer: a

Explanation: A new node is created with the required element. The pointer of the new node points the node to which the head node of the linked list is also pointing. The head node pointer is changed and it points to the new node which we created earlier. The entire process completes in O (1) time. Thus the asymptotic time complexity to insert an element in the second position of the linked list is O (1).

9)The concatenation of two list can performed in O(1) time. Which of the following variation of linked list can be used?

a) Singly linked list

b) Doubly linked list

c) Circular doubly linked list

d) Array implementation of list

Answer: c

Explanation: We can easily concatenate two lists in O (1) time using singly or doubly linked list, provided that we have a pointer to the last node at least one of the lists. But in case of circular doubly linked lists, we will break the link in both the lists and hook them together. Thus circular doubly linked list concatenates two lists in O (1) time.

10) What kind of linked list is best to answer question 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

Answer: d

Explanation: Arrays provide random access to elements by providing the index value within square brackets. In the linked list, we need to traverse through each element until we reach the nth position. Time taken to access an element represented in arrays is less than the singly, doubly and circular linked lists. Thus, array implementation is used to access the item at the position n.

11) Linked lists are not suitable to for the implementation of?

a) Insertion sort

b) Radix sort

c) Polynomial manipulation

d) Binary search

Answer: d

Explanation: It cannot be implemented using linked lists.

12) Linked list is considered as an example of _____ type of memory allocation.

a) Dynamic

b) Static

c) Compile time

d) Heap

Answer: a

Explanation: As memory is allocated at the run time.

13) In Linked List implementation, a node carries information regarding _____

a) Data

b) Link

c) Data and Link

d) Node

Answer: b

Explanation: A linked list is a collection of objects linked together by references from an object to another object. By convention these objects are names as nodes. Linked list consists of nodes where each node contains one or more data fields and a reference(link) to the next node.

14)Linked list data structure offers considerable saving in _____

a) Computational Time

b) Space Utilization

c) Space Utilization and Computational Time

d) Speed Utilization

Answer: c

Explanation: Linked lists saves both space and time.

15) 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

Explanation: Both Merge sort and Insertion sort can be used for linked lists. The slow random-access performance of a linked list makes other algorithms (such as quicksort) perform poorly, and others (such as heapsort) completely impossible. Since worst case time complexity of Merge Sort is $O(n\log n)$ and Insertion sort is $O(n^2)$, merge sort is preferred.

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

a) log2 n

b) $n/2$

c) log2 n – 1

d) n

Answer: d

Explanation: The worst-case happens if the required element is at last or the element is absent in the list. For this, we need to compare every element in the linked list. If n elements are there, n comparisons will happen in the worst case.

17)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)

Answer: d

Explanation: Depending on whether the array is full or not, the complexity in dynamic array varies. If you try to insert into an array which is not full, then the element is simply stored at the end, this takes O(1) time. If you try to insert into an array which is full, first you will have to allocate an array with double the size of the current array and then copy all the elements into it and finally insert the new element, this takes O(n) time.

18)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(n2)

Answer: b

Explanation: To count the number of elements, you have to traverse through the entire list, hence complexity is O(n).

19) 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)

Answer: a

Explanation: You need a temp variable to keep track of current node, hence the space complexity is O(1).

20) 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

Answer: b

Explanation: The pointer difference is calculated by taking XOR of pointer to previous node and pointer to the next node.

21) 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

Answer: c

Explanation: Generally, round robin fashion is employed to allocate CPU time to resources which makes use of the circular linked list data structure. Recursive function calls use stack data structure. Undo Operation in text editor uses doubly linked lists. Hash tables uses singly linked lists.

22)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

Answer: b

Explanation: Time complexity of inserting a new node at the head of the list is O(n) because you have to traverse through the list to find the tail node.

23) Which of the following real world scenarios would you associate with a stack data structure?

a) piling up of chairs one above the other

b) people standing in a line to be serviced at a counter

c) offer services based on the priority of the customer

d) tatkal Ticket Booking in IRCTC

Answer: a

Explanation: Stack follows Last In First Out (LIFO) policy. Piling up of chairs one above the other is based on LIFO, people standing in a line is a queue and if the service is based on priority, then it can be associated with a priority queue. Tatkal Ticket Booking Follows First in First Out Policy. People who click the book now first will enter the booking page first.

STACKS

1)What does 'stack underflow' refer to?

a) accessing item from an undefined stack

b) adding items to a full stack

c) removing items from an empty stack

d) index out of bounds exception

Answer: c

Explanation: Removing items from an empty stack is termed as stack underflow.

2)What is the time complexity of pop() operation when the stack is implemented using an array?

a) O(1)

b) O(n)

c) O(logn)

d) O(nlogn)

Answer: a

Explanation: pop() accesses only one end of the structure, and hence constant time.

3)Which of the following array position will be occupied by a new element being pushed for a stack of size N elements(capacity of stack > N).

a) S[N-1]

b) S[N]

c) S[1]

d) S[0]

Answer: b

Explanation: Elements are pushed at the end, hence N.

4)What happens when you pop from an empty stack while implementing using the Stack ADT in Java?

a) Undefined error

b) Compiler displays a warning

c) EmptyStackException is thrown

d) NoStackException is thrown

Answer: c

Explanation: The Stack ADT throws an EmptyStackException if the stack is empty and a pop() operation is tried on it.

5)Which of the following array element will return the top-of-the-stack-element for a stack of size N elements(capacity of stack > N).

a) S[N-1]

b) S[N]

c) S[N-2]

d) S[N+1]

Answer: a

Explanation: Array indexing start from 0, hence N-1 is the last index.

6)Which one of the following is an application of Stack Data Structure?

a)Managing function calls

b)The stock span problem

c)Arithmetic expression evaluation

d)All the above

Answer:A

7) The data structure required to check whether an expression contains balanced parenthesis is?

a) Stack

b) Queue

c) Array

d) Tree

Answer)a

8)The postfix form of A*B+C/D is?

a) *AB/CD+

b) AB*CD/+

c) A*BC+/D

d) ABCD+/*

Answer)B

9). Which data structure is needed to convert infix notation to postfix notation?

a) Branch

b) Tree

c) Queue

d) Stack

Answer)d

10)The prefix form of A-B/ (C * D ∧ E) is?

a) -/*∧ACBDE

b) -ABCD*∧DE

c) -A/B*C∧DE

d) -A/BC*∧DE

Answer)c

11)The result of evaluating the postfix expression 5, 4, 6, +, *, 4, 9, 3, /, +, * is?

a) 600

b) 350

c) 650

d) 588

Answer)b

12)Consider the linked list implementation of a stack. Which of the following node is considered as Top of the stack?

a) First node

b) Last node

c) Any node

d) Middle node

Answer)a

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

Push(1);

Pop();

Push(2);

Push(3);

Pop();

Push(4);

Pop();

Pop();

Push(5);

After the completion of all operation, the no of element present on stack are

a) 1

b) 2

c) 3

d) 4

Answer)a

14)Which of the following is not an inherent application of stack?

a) Reversing a string

b) Evaluation of postfix expression

c) Implementation of recursion

d) Job scheduling

Answer)d

15) Which of the following operation take worst case linear time in the array implementation of stack?

a) Push

b) Pop

c) IsEmpty

d) None

Answer)d

16) Which of the following application generally use a stack?

a) Parenthesis balancing program

b) Syntax analyzer in compiler

c) Keeping track of local variables at run time

d) All of the above

Answer)d

17) What is the minimum number of stacks of size n required to implement a queue of size n?

a) One

b) Two

c) Three

d) Four

Answer)b

18) Consider the usual implementation of parentheses balancing program using stack. What is the maximum number of parentheses that will appear on stack at any instance of time during the analysis of (( ) (( )) (( )))?

a) 1

b) 2

c) 3

d) 4

Answer)c

QUEUES

1) 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 ?

a) Queue

b) Stack

c) Tree

d) Linked list

Answer)a

2)The data structure required for Breadth First Traversal on a graph is?

a) Stack

b) Array

c) Queue

d) Tree

Answer)c

3). A queue follows _____

a) FIFO (First In First Out) principle

b) LIFO (Last In First Out) principle

c) Ordered array

d) Linear tree

Answer)a

4)Circular Queue is also known as _____

a) Ring Buffer

b) Square Buffer

c) Rectangle Buffer

d) Curve Buffer

Answer)a

5)A data structure in which elements can be inserted or deleted at/from both the ends but not in the middle is?

a) Queue

b) Circular queue

c) Dequeue

d) Priority queue

Answer)c

6) 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

Answer: a

Explanation: When Rear = MAX_SIZE – 1, there will be no space left for the elements to be added in queue. Thus queue becomes full.

7)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

Answer)c

8)Which of the following is not the type of queue?

a) Ordinary queue

b) Single ended queue

c) Circular queue

d) Priority queue

Answer: b

Explanation: Queue always has two ends. So, single ended queue is not the type of queue.

9)In Breadth First Search of Graph, which of the following data structure is used?

a) Stack

b) Queue

c) Linked list

d) None

Answer)b

10). In linked list implementation of a queue, where does a new element be inserted?

a) At the head of link list

b) At the tail of the link list

c) At the centre position in the link list

d) None

Answer)b

11). In the array implementation of circular queue, which of the following operation take worst case linear time?

a) Insertion

b) Deletion

c) To empty a queue

d) None

Answer)d

12)In linked list implementation of queue, if only front pointer is maintained, which of the following operation take worst case linear time?

a) Insertion

b) Deletion

c) To empty a queue

d) Both a) and c)

Answer)d

13) A circular queue is implemented using an array of size 10. The array index starts with 0, front is 6, and rear is 9. The insertion of next element takes place at the array index.

a) 0

b) 7

c) 9

d) 10

Answer)a

14)n linked list implementation of a queue, front and rear pointers are tracked. Which of these pointers will change during an insertion into a NONEMPTY queue?

a) Only front pointer

b) Only rear pointer

c) Both front and rear pointer

d) None of the front and rear pointer

Answer)b

15) In linked list implementation of a queue, front and rear pointers are tracked. Which of these pointers will change during an insertion into EMPTY queue?

a) Only front pointer

b) Only rear pointer

c) Both front and rear pointer

d) None

Answer)c

16)An array of size MAX_SIZE is used to implement a circular queue. Front, Rear, and count are tracked. Suppose front is 0 and rear is MAX_SIZE -1. How many elements are present in the queue?

a) Zero

b) One

c) MAX_SIZE-1

d) MAX_SIZE

Answer)d

17)What is the worst case time complexity of a sequence of n queue operations on an initially empty queue?

a) θ (n)

b) θ (n + k)

c) θ (nk)

d) θ (n2)

Answer)a

TREES

1)The height of a BST is given as h. Consider the height of the tree as the no. of edges in the longest path from root to the leaf. The maximum no. of nodes possible in the tree is?

a) 2^{h-1} -1

b) 2^{h+1} -1

c) 2^{h} +1

d) 2^{h-1} +1

Answer)a

2)The no of external nodes in a full binary tree with n internal nodes is?

a) n

b) n+1

c) 2n

d) 2n + 1

Answer)b

3)The difference between the external path length and the internal path length of a binary tree with n internal nodes is?

a) 1

b) n

c) n + 1

d) 2n

Answer)d

4) Suppose a binary tree is constructed with n nodes, such that each node has exactly either zero or two children. The maximum height of the tree will be?

a) (n+1)/2

b) (n-1)/2

c) n/2 -1

d) (n+1)/2 -1

Answer)b

5)Which of the following statement about binary tree is CORRECT?

a) Every binary tree is either complete or full

b) Every complete binary tree is also a full binary tree

c) Every full binary tree is also a complete binary tree

d) A binary tree cannot be both complete and full

Answer)c

6)Suppose we have numbers between 1 and 1000 in a binary search tree and want to search for the number 363. Which of the following sequence could not be the sequence of the node examined?

a) 2, 252, 401, 398, 330, 344, 397, 363

b) 924, 220, 911, 244, 898, 258, 362, 363

c) 925, 202, 911, 240, 912, 245, 258, 363

d) 2, 399, 387, 219, 266, 382, 381, 278, 363

Answer)C

7) In full binary search tree every internal node has exactly two children. If there are 100 leaf nodes in the tree, how many internal nodes are there in the tree?

a) 25

b) 49

c) 99

d) 101

Answer)c

8)Which type of traversal of binary search tree outputs the value in sorted order?

a) Pre-order

b) In-order

c) Post-order

d) None

Answer)b

9) Suppose a complete binary tree has height h>0. The minimum no of leaf nodes possible in term of h is?

a) $2^h - 1$

b) $2^{h-1} + 1$

c) 2^{h-1}

d) $2^h + 1$

Answer)c

10) If a node having two children is to be deleted from binary search tree, it is replaced by its

a) In-order predecessor

b) In-order successor

c) Pre-order predecessor

d) None

Answer)b

11) A binary search tree is formed from the sequence 6, 9, 1, 2, 7, 14, 12, 3, 8, 18. The minimum number of nodes required to be added in to this tree to form an extended binary tree is?

a) 3

b) 6

c) 8

d) 11

Answer)d

12) In a full binary tree, every internal node has exactly two children. A full binary tree with 2n+1 nodes contains

a) n leaf node

b) n internal nodes

c) n-1 leaf nodes

d) n-1 internal nodes

Answer)b

13) the run time for traversing all the nodes of a binary search tree with n nodes and printing them in an order is

a) O(nlg(n))

b) O(n)

c) O($\sqrt{n}$)

d) O(log(n))

Answer)b

14)When a binary tree is converted in to an extended binary tree, all the nodes of a binary tree in the external node becomes

a) Internal nodes

b) External nodes

c) Root nodes

d) None

Answer)a

15)If n numbers are to be sorted in ascending order in O(nlogn) time, which of the following tree can be used

a) Binary tree

b) Binary search tree

c) Max-heap

d) Min-heap

Answer)d

16) If n elements are sorted in a binary search tree. What would be the asymptotic complexity to search a key in the tree?

a) O(1)

b) O(logn)

c) O(n)

d) O(nlogn)

Answer)c

17) If n elements are sorted in a balanced binary search tree. What would be the asymptotic complexity to search a key in the tree?

a) O(1)

b) O(logn)

c) O(n)

d) O(nlogn)

Answer)b

18)The minimum number of elements in a heap of height h is

a) 2^{h+1}

b) 2^h

c) $2^h -1$

d) 2^{h-1}

Answer)b

19)The maximum number of elements in a heap of height h is

a) $2^{h+1} -1$

b) 2^h

c) $2^h -1$

d) 2^{h-1}

Answer)a

20) A threaded binary tree is a binary tree in which every node that does not have right child has a thread to its

a) Pre-order successor

b) In-order successor

c) In-order predecessor

d) Post-order successor

Answer)b

21)In which of the following tree, parent node has a key value greater than or equal to the key value of both of its children?

a) Binary search tree

b) Threaded binary tree

c) Complete binary tree

d) Max-heap

Answer)d

22)A binary tree T has n leaf nodes. The number of nodes of degree 2 in T is

a) $\log_2 n$

b) n-1

c) n

d) 2^n

Answer)b

23)A binary search tree is generated by inserting in order the following integers:

50, 15, 62, 5, 20, 58, 91, 3, 8, 37, 60, 24

The number of the node in the left sub-tree and right sub-tree of the root, respectively, is

a) (4, 7)

b) (7, 4)

c) (8, 3)

d) (3, 8)

Answer)b

SORTING ALGORITHM:

Table 1: Complexities and stability of some sorting algorithms

Name of the algorithm	Average case time complexity	Worst case time complexity	Stable?
Bubble sort	$\Theta(n^2)$	$O(n^2)$	Yes
Selection sort	$\Theta(n^2)$	$O(n^2)$	No
Insertion sort	$\Theta(n^2)$	$O(n^2)$	Yes
Merge sort	$\Theta(n \log_2 n)$	$O(n \log_2 n)$	Yes
Quick sort	$\Theta(n \log_2 n)$	$O(n^2)$	No
Bucket sort	$\Theta(d(n+k))$	$O(n^2)$	Yes
Heap sort	$\Theta(n \log_2 n)$	$O(n \log_2 n)$	No

1) Which of the following is not a stable sorting algorithm?

a) Insertion sort

b) Selection sort

c) Bubble sort

d) Merge sort

Answer)b

2)Which of the following is a stable sorting algorithm?

a) Merge sort

b) Typical in-place quick sort

c) Heap sort

d) Selection sort

Answer)a

3)Which of the following is not an in-place sorting algorithm?

a) Selection sort

b) Heap sort

c) Quick sort

d) Merge sort

Answer)d

4) Running merge sort on an array of size n which is already sorted is

a) O(n)

b) O(nlogn)

c) O(n^2)

d) None

Answer)b

5)The time complexity of a quick sort algorithm which makes use of median, found by an O(n) algorithm, as pivot element is

a) O(n^2)

b) O(nlogn)

c) O(nloglogn)

d) O(n)

Answer)b

6)Which of the following is not a noncomparison sort?

a) Counting sort

b) Bucket sort

c) Radix sort

d) Shell sort

Answer)d

7) The time complexity of heap sort in worst case is

a) O(logn)

b) O(n)

c) O(nlogn)

d) O(n²)

Answer)c

8) If the given input array is sorted or nearly sorted, which of the following algorithm gives the best performance?

a) Insertion sort

b) Selection sort

c) Quick sort

d) Merge sort

Answer)a

9)Which of the following algorithm pays the least attention to the ordering of the elements in the input list?

a) Insertion sort

b) Selection sort

c) Quick sort

d) None

Answer)b

10)Consider the situation in which assignment operation is very costly. Which of the following sorting algorithm should be performed so that the number of assignment operations is minimized in general?

a) Insertion sort

b) Selection sort

c) Heap sort

d) None

Answer)b

11). Time complexity of bubble sort in best case is

a) θ (n)

b) θ (nlogn)

c) θ (n2)

d) θ (n(logn) 2)

Answer)A

12). Given a number of elements in the range [0….n^3]. which of the following sorting algorithms can sort them in O(n) time?

a) Counting sort

b) Bucket sort

c) Radix sort

d) Quick sort

Answer)c

13)Which of the following algorithms has lowest worst case time complexity?

a) Insertion sort

b) Selection sort

c) Quick sort

d) Heap sort

Answer)d

14) Which of the following sorting algorithms is/are stable

a) Counting sort

b) Bucket sort

c) Radix sort

d) All of the above

Answer)d

15) Counting sort performs …………. Numbers of comparisons between input elements.

a) 0

b) n

c) nlogn

d) n^2

Answer)A

16). The radix sort does not work correctly if each individual digit is sorted using

a) Insertion sort

b) Counting sort

c) Selection sort

d) Bubble sort

Answer)c

17) Which of the following sorting algorithm has the running time that is least dependant on the initial ordering of the input?

a) Insertion sort

b) Quick sort

c) Merge sort

d) Selection sort

Answer)d

18) Time complexity to sort elements of binary search tree is

a) O(n)

b) O(nlogn)

c) $O(n^2)$

d) $O(n^2logn)$

Answer)a

19)The lower bound on the number of comparisons performed by comparison-based sorting algorithm is

a) Ω (1)

b) Ω (n)

c) Ω (nlogn)

d) Ω (n^2)

Answer)c

20)Which of the following algorithm design technique is used in the quick sort algorithm?

a) Dynamic programming

b) Backtracking

c) Divide-and-conquer

d) Greedy method

Answer)C

21)Merge sort uses

a) Divide-and-conquer

b) Backtracking

c) Heuristic approach

d) Greedy approach

Answer)a

22) For merging two sorted lists of size m and n into sorted list of size m+n, we require comparisons of

a) O(m)

b) O(n)

c) O(m+n)

d) O(logm + logn)

Answer)c

23)A sorting technique is called stable if it

a) Takes O(nlogn) times

b) Maintains the relative order of occurrence of non-distinct elements

c) Uses divide-and-conquer paradigm

d) Takes O(n) space

Answer)b

24) In a binary max heap containing n numbers, the smallest element can be found in time

a) θ (n)

b) θ (logn)

c) θ (loglogn)

d) θ (1)

Answer)a

GRAPHS

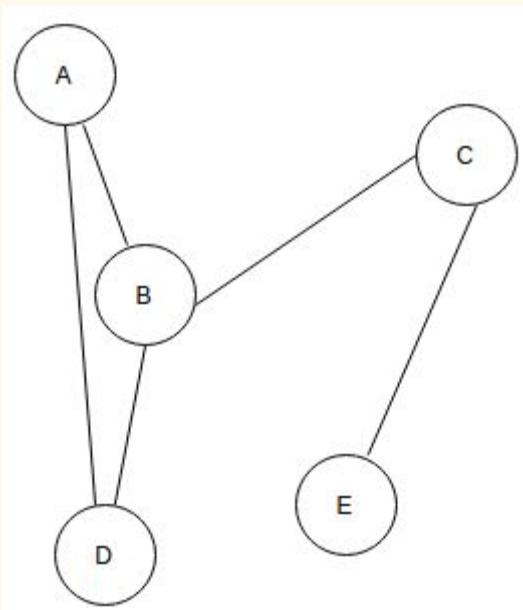**1)Which of the following statements for a simple graph is correct?**

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

Answer: a

Explanation: In a walk if the vertices are distinct it is called a path, whereas if the edges are distinct it is called a trail.
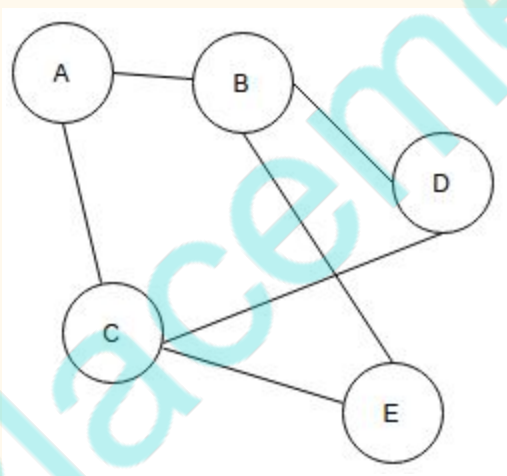
2) In the given graph identify the cut vertices.

a) B and E

b) C and D

c) A and E

d) C and B

Answer)d

3)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

Answer)c

4) What is the number of edges present in a complete graph having n vertices?
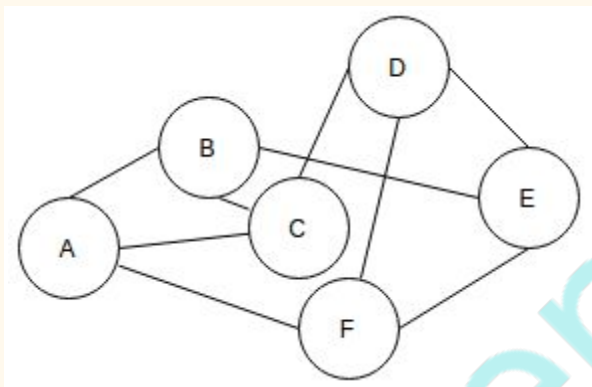
a) (n*(n+1))/2

b) (n*(n-1))/2

c) n

d) Information given is insufficient

Answer)b

5) The given Graph is regular.

a) True

b) False

Answer: a

Explanation: In a regular graph, degrees of all the vertices are equal. In the given graph the degree of every vertex is 3

6) In a simple graph, the number of edges is equal to twice the sum of the degrees of the vertices.

a) True

b) False

Answer: b

Explanation: The sum of the degrees of the vertices is equal to twice the number of edges.

7) A connected planar graph having 6 vertices, 7 edges contains _____ regions.

a) 15

b) 3

c) 1

d) 11

Answer: b

Explanation: By euler's formula the relation between vertices(n), edges(q) and regions(r) is given by n-q+r=2.

8)What is the maximum number of edges in a bipartite graph having 10 vertices?

a) 24

b) 21

c) 25

d) 16

Answer: c

Explanation: Let one set have n vertices another set would contain 10-n vertices.

Total number of edges would be n*(10-n), differentiating with respect to n, would yield the answer.

9) Which of the following is true?

a) A graph may contain no edges and many vertices

b) A graph may contain many edges and no vertices

c) A graph may contain no edges and no vertices

d) A graph may contain no vertices and many edges

Answer: b

Explanation: A graph must contain at least one vertex.

10) For a given graph G having v vertices and e edges which is connected and has no cycles, which of the following statements is true?

a) v=e

b) v = e+1

c) v + 1 = e

d) v = e-1

Answer: b

Explanation: For any connected graph with no cycles the equation holds true.

11). A graph with all vertices having equal degree is known as a _____

a) Multi Graph

b) Regular Graph

c) Simple Graph

d) Complete Graph

Answer: b

Explanation: The given statement is the definition of regular graphs.