Data Structure and Algorithms

Paper code –PCC-CS-301

3rd Semester CSE

PART - A

(Multiple Choice Type Questions)

Module – I

Linear Data Structure [Array]

1. The memory address of the first element of an array is called
   a) Floor address
   b) Foundation address
   c) First address
   d) Base address
2. Each element of an array arr [20][50] requires 4 bytes of storage. Base address of arr is 2000. The location of arr[10][10] when the array is stored as column major fashion is
   a) 2820 b) 2840 c) 4048 d) 4840
3. The expression that accesses the (i, j) th entry (i=0,1,…m-1,j=0,1,…n-1 of an m×n matrix (stored in column major order) is
   a) n × (i-1)+j b) m×(j-1)+I c) n × (i-1)+ (j-1) d) m×(j-1)+ (i-1)
4. In C language arrays are stored in which representation?
   a) Column major b)Row major c)Layer major d)None of these
5. Base address of a floating point 2D array a is 2000.a is stored in row- major order in memory, lower limit is adapted as 0 and the dimension of a is 4 and 5 respectively. What will be the address of A[2][3] ?
   a) 2022 b)2052 c)2026 d)2044
6. Dynamic memory allocation use
   a) calloc b) malloc c) free d)all of these
7. In C language malloc () returns
   a) Integer pointer b) null pointer c) float pointer d)void pointer these
8. The expression that accesses the (i,j) th entry of an m × n matrix(stored in column major order) is
   a) n ×(i-1) +j b) m × (j-1)+1 c) n × (i -1)+(j-1) d) m × (j-1)+(i-1)
9. An abstract data type (ADT) is type (or class) for objects whose behavior is defined by a set of values and a set of operations
   a) Same as an abstract class
   b) A data type that cannot be instantiated
   c) A data type for which only the operations defined on it can be used but none else
   d) None of the above
10. A matrix “a” is called lower triangular if and only if for all j>i, a[i,j]=0. If such a matrix is to be stored in a one dimensional array, A, then a[i, j] could be mapped to the following index of A:
    a) ½*i (i+1) +j b) i + j c) i*(i+1) + j d) None of the these
11. For a strictly lower triangular matrix the element a_{ij}, where i is the row and j is the column position respectively, is 0 for
    a) i≠ j i>=j b) i<= j c) i> j d) i< j

Module – II/ Linear Data Structure [Stack and Queue]

12. The following sequence of operations is performed on a stack: push(1,) push(2),pop, push(1), push(2), pop, pop, pop, push(2),pop. The sequence of popped out values are
    a) 2,2,1,2,1 b) 2,2,1,1,2 c) 2,1,2,2,1 d) 2,1,2,2,2

13. A stack is implemented using an array with the following declaration:
    int stack [100] ; int stacktop=0;
    To perform the POP operation, which of the following is correct?
    a) x=stack [stacktop++] b) x= stack [++stacktop]
    c) x=stack [stacktop--] d) x= stack [--stacktop]

14. Stack is used in
    a) Recursion b) Invoking functions c) All of the above d) None of the above

15. The other name for prefix notation is
    a) Reverse polish b) Polish c) Infix d) None of the above

16. Stack can be implemented using
    a) Arrays b) Linked lists c) both a & b d) None of the above

17. In Stack data structure, insertions and deletions are made at
    a) both ends b) one end only c) middle of the stack d) None of the these

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

19. The operation for adding an entry to a stack is traditionally called:
    a) Add b) Append c) Insert d) Push

20. The operation for removing an entry from a stack is traditionally called:
    a) Delete b) Peek c) Pop d) Remove

21. Which of the following stack operations could result in stack underflow?
    a) Is_empty b) Pop c) Push d) Two or more of the above answers.

22. The postfix equivalent of the prefix *+ab-cd is
    a) AB+CD-* b) abcd+-* c) ab+cd*- d) ab+-cd*

23. A postfix expression for the infix expression a+b*(c+d)/f+d*e
    a) ab+cd+*f/d+e* b) abcd+*f/+de*+ c) a*b+cd/f*de++ d) None of these

24. A prefix expression for the infix expression a*(b+c)/e-f
    a)/*a+bc-ef b) -/*+abcef c) -/*a+bcef d) None of these

25. Translating the infix expression(P) into postfix notation, we get
    [P = A + (B∗ C – (D / (E + F)) ∗ G) ∗ H]
    a) ABC ∗ DEF / + G ∗– H ∗ +
    b) ABC ∗ + DEF + / – G ∗ H – +
    c) ABC ∗ DEF + / G ∗– H ∗ +
    d) None of these

26. Which data structure is needed to convert infix notations to postfix notations?
    a) Linear list b) Queue c) Tree d) Stack

27. Stack is ……
a) a linear data structure b) a nonlinear data structure c) None of the above

28. Stack is also called as
a) FIFO b) LIFO c) both (i) and (ii) d) None of the above

29. The number of elements that can be removed from the stack at any time is ……
a) 3 b)4 c)1 d) 0

30. Suppose we have an array implementation of the stack structure, with ten items in the stack stored at data[0] through data[9]. The CAPACITY is 42. Where does the push method place the new entry in the array?
a) data[0] b) data[1] c) data[9] d) data[10]

31. Which data structure is used in evaluating mathematical expressions with parentheses?
a)Stack b) Queue c) Tree d) Graph

32. Which of the following applications may use a stack?
a) A parentheses balancing program. b) Keeping track of local variables at run time.
c) Syntax analyzer for a compiler. d) All of the above.

33. The five items A, B, C, D, E are pushed in a stack, one after the other starting from A. The stack is popped four times and each element is inserted in a queue. Then two elements are deleted from the queue and pushed back on the stack. Now one item is popped from the stack. The popped item is
a) A b) B c) C d) D

34. Suppose you opened a notepad, a music player, an excel sheet, and also you are doing your data structure programming simultaneously. Your OS implements which data structure for it.
a) Stack b) Queue c) Tree d) Linked List

35. Simulations are implemented using _____
a) Stack b) Queue c) Linked List d) Tree

36. The evaluation of the postfix expression 3,5,7,*,+,12,% is
a)2 b)3 c)0 d)3.17

37. If we evaluate the following postfix expression 23 5 7 * - 12 +
The result will be
a) 12 b) 0 c) -12 d) 35

38. The number of stacks required to implement mutual recursion is
a) 3 b)2 c)1 d)None of these

39. Insertion in stack is done in
a) front b) rear c) top d) bottom

40. The integers 1,2,3,4 are pushed into the stack in that order. They may be popped out of the stack in any valid order. The integers, which are popped out produce a permutation of the numbers 1, 2, 3, 4.Which of the following permutation can never be produced in such a way?
a) 1,2,3,4 b) 4,2,3,1 c) 4,3,2,1 d) 3,2,4,1

41. Which data structure is used to manage Printer Buffer?
a) Stack b) Queue c) Linked List d) Tree

42. Which of the following is essential for converting an infix expression to postfix notation?
a) A parse tree b) An operand stack
c) An operator stack d)None of these

43. Ascending priority queue is one in which the item removed is
a) the smallest item b) the largest item c) any item

44. Descending priority queue is one in which the item removed is
a) the smallest item b) the largest item c) any item

45. One difference between a queue and a stack is:
a) Queues require linked lists, but stacks do not.
b) Stacks require linked lists, but queues do not.
c) Queues use two ends of the structure; stacks use only one.
d) Stacks use two ends of the structure, queues use only one.

46. If the characters 'D', 'C', 'B', 'A' are placed in a queue (in that order), and then removed one at a time, in what order will they be removed?
a) ABCD b) ABDC c) DCAB d) DCBA

47. Let P be the queue of integers defined as follows

```
#define MAX 50
struct queue {
        int items[MAX];
        int front, rear;
}q;
```

To insert an elements in the queue we can use
a)++q.items[q.rear]=x; b) q.items[++q.rear]=x; c) q.items[++q.rear]++=x;

48. The deque can be used
a) as a stack b) as a queue c) both as a stack and as a queue d) None of the above

49. I) In the case of priority queue elements can be inserted
a) at the ends b) at any position c) a and b d) None of the above

50. Priority queue can be implemented using
a)Array b)Linked list c)Heap d)All of these

51. Suppose we have a circular array implementation of the queue structure, with ten items in the queue stored at data[2] through data[11]. The current capacity is 42. Where does the insert method place the new entry in the array?
a) data[1] b) data[2] c) data[11] d) data[12]

52. If data is a circular array of CAPACITY elements, and rear is an index into that array, what is the formula for the index after rear?
a) (rear % 1) + CAPACITY b) rear % (1 + CAPACITY)
c) (rear + 1) % CAPACITY d) rear + (1 % CAPACITY)

53. Suppose getFront is called on a priority queue that has exactly two entries with equal priority. How is the return value of getFront selected?
a) One is chosen at random b) The one which was inserted first.
c) The one which was inserted most recently. d) This can never happen (violates the precondition)

54. Queue is implemented with a linked list, keeping track of a front node and a rear node with two reference variables. Which of these reference variables will change during an insertion into a NONEMPTY queue?
a) Neither changes b) Only front changes c) Only rear changes. d) Both change

55. Which of the following data structure may give overflow error, even though the current number of elements in it, is less than its size.
a) Simple queue b) Circular queue c) Stack d) None of the above.

56. A linear list in which elements can be added or removed at either end but not in the middle is known as

a)Stack b) Queue c) Dequeue d) Heap

57. The initial configuration of queue a,b,c,d('a' is at the front).To get the configuration d,c,b,a One needs a minimum of

a) 2 deletions and 3 additions
b) 3 deletions and 2 additions
c) 3 deletions and 3 additions
d) 3 deletions and 4 additions

58. The rear and front end of a linear queue is used for

a) deletion, insertion
b) searching, sorting
c) insertion deletion
d) none of these

59. Queue is implemented with a circular array, keeping track of front, rear, and manyItems (the number of items in the array). Suppose front is zero, and rear is one less than the current capacity. What can you tell me about many Items?

a) manyItems must be zero.
b) manyItems must be equal to the current capacity.
c) count could be zero or the capacity, but no other values could occur.
d) None of the above.

Module – III/ Recursion and Linked List

60. The list data structure can be defined recursively

a)True b) False

61. In Linked list, the logical order of elements

a) Is same as their physical arrangement
b) Is determined by their physical arrangement
c) Is not necessarily equivalent to their physical arrangement
d) None of these

62. Linked List are not suitable for

a)Stack b)Dequeue c)AVL Tree d)Binary Search

63. Linked List are not suitable data structure for which one of the following problems?

a) Insertion sort b) Radix sort c) Binary search d) Polynomial addition

64. Which is the correct notation to delete the last node p from a doubly linked list ?(prev is the pointer to the previous node and next is the pointer pointing to the next node)

a) p->next=NULL b) p=NULL c) p->prev->next=NULL
d) None of these

65. Inserting a new node after a given node in a doubly Linked list requires

a) four pointer exchanges
b) two pointer exchanges
c) one pointer exchanges
d) no pointer exchanges

66. The Ackerman function, for all non-negative values of m and n is recursively defined as

$$A(m,n) = \begin{cases} n+1 & \text{if } m = 0 \\ A(m-1,1) & \text{if } m \neq 0 \text{ but } n = 0 \\ A(m-1, A(m,n-1)) & \text{if } m \neq 0 \text{ and } n \neq 0 \end{cases}$$

Therefore the value of A(1,2) is

a) 4 b) 3 c) 5 d) 2

67. Recursion may be implemented by

a) linked-list b) stack c) queue d) dequeue

68. Fibonacci function fib(n)=fib(n-1)+ fib(n-2) is an example of

a) Linear recursion b) Binary recursion

c) Non-linear recursion d) Mutual recursion

69. In a circularly linked list organization, insertion of a record involves the modification of

a) no pointer b) 1 pointer c) 2 pointers d) 3 pointers

70. Underflow condition in linked list may occur when attempting to :

a) insert a new node when there is free space for it

b) delete a nonexistent node in the list

c) delete a node in empty list

d) none of these

71. A technique, which collects all deleted space onto free storage list, is called-

a) Static memory

b) Garbage collection allocation

c) Dynamic allocation

d) None of the above

72. Which of the following statements is not true?

a) Each time a function calls itself, it must be nearer in some sense to a solution

b) Recursive function are always fast and use less memory

c) When the last statement of function is a recursive call then it is known as tail recursion

d) In implementing recursion stacks are generally used.

73. Choose the statement to be inserted in the following function for insertion of an element into a doubly linked list :

```
structdlist
{
        intval;
        structdlist *next;
        structdlist *prev;
};
void insert(structdlist *x, structdlist *prev)
{
    /*x contains address of the new element */
    x->next=prev->next;
    x->prev=prev;
    /*statements to be inserted here*/
}
```

a) prev->prev=x->next->next=x;

b) prev->next=x->next->next=x;

c) prev->prev=x->next->prev=x;

d) prev->next=x->next->prev=x;

Module – IV/ Nonlinear Data Structure [Tree]

74. Which of the following process is faster for threaded trees compared with their unthreaded counterparts

a) Insertions b) Deletion c) Traversal d) Searching

75. Which of the following traversal techniques lists the elements of a binary search tree in ascending order?

a) Pre-order b) Post-order c) In order d)None of these

76. The depth of a complete binary tree with n nodes
    a) log(n+1)-1 b) log(n) c) log(n-1)+1 d)log(n)+1
77. In a binary search tree, if the number of nodes of a tree is 9, then the minimum height of the tree is
    a) 9 b) 5 c) 4 d) none of these
78. Number of possible binary trees with 3 node is
    a) 3 b) 2 c) 4 d) 5
79. Total nodes in a 2-tree(Strictly binary tree) with thirty leaves are
    a) 60 b) 58 c) 59 d) 57
80. In a height balanced tree, heights of two sub –trees of every node differ by more than
    a) 2 b) 0 c) 1 d) -1
81. How many BST can be formed with 1,2,3,4?
    a) 1 b) 2 c) 4 d) 6
82. A full binary tree with n non-leaf nodes contains
    a) $\log_2$ (n) nodes b) n+1 nodes c) 2n nodes d) 2n+1 nodes
83. A full binary tree with n leaves contains
    a) n nodes b) log n nodes c) 2n-1 nodes d) 2^n nodes
84. 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 nodes in the left subtree and right subtree of the root respectively is
    a) (4,7) b) (7,4) c) (8,3) d) (3,8)
85. If a binary tree is threaded for in order traversal a right NULL link of any node is replaced by the address of its
    a) successor b) predecessor c) root d) own
86. In a max heap, both the addition and deletion operations can be performed in time
    a) O(log n) b) O(nlog n) c) O(n) d) $O(n^2)$
87. In a Heap,the right child of a node in position 10 will be in the position of
    a) 20 b) 21 c) 9 d) 5
88. A binary tree may be reconstructed from
    a) inorder traversal sequence only
    b) preorder traversal sequence only
    c) postorder traversal sequence only
    d) both inorder and preorder traversal sequence
89. Maximum possible height of an AVL tree with 7 nodes is
    a) 3 b) 4 c) 5 d) 6
90. In traversing non-empty binary tree, visit the root is made in the last in
    a) Preorder traversal
    b) Inorder traversal
    c) Postorder traversal
    d) None of these
91. The in-order and post-order traversal of a binary tree are DBEAFC and DEBFCA respectively. What will be the total number of nodes in the left subtree of the given tree?
    a) 1 b) 4 c) 5 d) none of these
92. A B-tree is
    a) Always balanced b) an ordered tree
    c) A directed tree d) all of these
93. A Binary tree is a type of tree
    a) That is ordered
    b) b) such that no node has degree more than 2

c) For which both (a) and (b) above are correct
d) in which non-leaf nodes will have degree 2

94. In array representation of Binary tree, if the index number of a child node is 6 then the index number of its parent node is
a) 2 b) 3 c) 4 d) 5

95. In an AVL tree the balancing is needed when balancing factor of any node becomes
a) 1 or -1 b) 0 or -1 c) -2 or 2 d) -1 or 0

96. The values in a BST can be sorted in ascending order by using which of the following traversals?
a) Pre-order b) In-order c) Post-order d) Level-order

97. Which tree structure is used for efficient access of records residing in disc memory?
b) AVL tree b) B tree c) 2-3 tree d) Binary tree

Module – V/ Nonlinear Data Structure [Graph]

98. A complete directed graph of 5 nodes has…… number of edges
a) 5 b) 10 c) 20 d) 25

99. Breadth-first-search algorithm uses ……data structure
a) Stack b) Queue c) Binary tree d) none of these

100. Adjacency matrix of a digraph is
a) Identity matrix b) Symmetric matrix c) asymmetric matrix d) none of these

101. A graph G with n nodes is bipartite if it contains
a) n edges b) a cycle of odd length c) no cycle of odd length d) n^2 edges

102. The vertex, removal of which makes a graph disconnected is called
a) Pendant vertex
b) Bridge
c) Articulation point
d) Colored vertex

103. A vertex of in -degree zero in a directed graph is called
a) Articulation point b) sinkc) isolated vertex d) rootvertex

104. BFS
a) Scans all incident edges before moving to the other vertex
b) Scans adjacent unvisited vertex as soon as possible
c) Is same as backtracking
d) None of these

105. A non-planer graph with minimum number of vertices has
a) 9 edges, 6 vertices b) 6 edges, 4 vertices
c) 10 edges, 5 vertices d) 9 edges, 5 vertices

106. Maximum number of edges in a n-node undirected graph without self loop is
a) n^2 b) n(n-1)/2
c) n-2 d) (n+1)n/2

107. BFS constructs
a) a minimal cost spanning tree of a graph
b) a depth first spanning tree of a graph
c) a breadth first spanning tree of a graph
d) none of these

108. A vertex of with degree one in a graph is called
a) Leaf b) Pendant vertex c) End vertex d) none of these

109. Which method of traversal does not use stack to hold nodes that are waiting to be processed?

a) Breadth- first
b) Depth-first
c) D-search
d) None of these

110. Which data structure is used for depth first traversal of a graph?
a) Array
b) Linked list
c) Stack
d) Queue

111. A simple undirected graph with eight (8) vertices is said to be completed if number of edges equal to
a) 56 b) 28 c) 16 d) 24

112. Any connected graph with x vertices must have at least
a) x+1 edges b)x-1 edges c) x edges d) x/2 edges

113. The element at the root of the heap is
a) largest b) smallest c) may be largest or smallest d) none of these

114. A digraph in which , outdegree is same as indegree is called
a) Balanced b) symmetric c) regular d) complete

Module – VI/Searching, Sorting and Time –Space Complexity

115. In external sorting technique all data reside in
a) Primary memory b) Secondary memory c) both (a) and (b)
d) None of these

116. Which one is the best time among the following algorithms?
a) O(n) b) $O(\log_2 n)$ c)$O(2^n)$d) $O(n\log_2 n)$

117. Average case time complexity of quick sort
a) $O(N \log_2 N)$b) $O(N \log N)$ c) $O(N^2)$ d)$O(N^3)$

118. If $f(n)=1000 n\log n +500n^4 +0.52^n$ then f(n) is
a) $O(n^4)$ b) O(nlog n) c) $O(2^n)$d) None of these

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

120. Selection sort and Quick sort both fall into the same category of sorting algorithms. What is this category?
a) O(nlogn) sorts b) Divide-and-Conquer Sorts c) Interchange sorts d) Average time is quadratic

121. What is the Big Oh notation of the following expression $F(n)=n \log n^2 + n^2 + e^{\log n}$
a) O(n) b) $O(n^2)$ c) $O(n \log n^2)$ d) $O(e^{\log n})$

122. A sort which compares adjacent elements in a list and switches where necessary is
a) Insertion sort b) Heap sort c) Quick sort d) Bubble sort

123. Which of the following sorting techniques requires extra space, than the data to be sorted?
a) Selection sort b) Heap sort c) Bubble sort d) None of these

124. Stability of Sorting Algorithm is important for
a) Sorting records on the basis of multiple keys
b) Worst case performance of sorting algorithm
c) Sorting alpha numeric keys as they are likely to be the same
d) None of these

125. The worst case complexity of binary search is
a) O(log n) b) O(nlog n) c) O(n) d) $O(n^2)$

126. Which of the following sorting procedures is the slowest?
a) Quick sort b) Heap sort c) Merge sort d) Bubble sort

127. The best case time complexity of bubble sort technique is
a) O(n) b) $O(n^2)$ c) O(nlog n) d) O(log n)

128. Four algo do the same task. Which algo should execute the slowest for large values of n?
a) $O(n^2)$ b) O(n) c) $O(\log_2 n)$ d) $O(2^n)$

129. What will be the time complexity for selection sort for an array of n elements?
a) O(log n) b) O(nlog n) c) O(n) d) $O(n^2)$

130. A machine needs a minimum of 100 sec to sort 1000 names by quick sort. The minimum time needed to sort 100 names will be approximately
a) 72.7 sec b) 11.2 sec c) 50.2 sec d) 6.7 sec

131. Which of the following algorithm should execute the slowest for large values of N?
a) O(N) b) $O(N^2)$ c) $9 (\log_2 N)$ d) None of these

132. Best case time complexity of insertion sort is
a) O(1) b) O(n)c) O(nlog n) d) $O(n^2)$

133. A sort, which iteratively passes through a list to exchange the first element with any element less than it and then repeats with a new first element is called
a) Bubble sort b) Quick sort c) Heap sort
d) Selection sort

134. Which of the following shows the correct relationship among some of the more common computing times for algorithm?
a) $O(\log n) < O(n) < O(n*\log n) < O(2^n) < O(n^2)$
b) $O(n) < O(\log n) < O(n*\log n) < O(2^n) < O(n^2)$
c) $O(n) < O(\log n) < O(n*\log n)) < O(n^2) < O(2^n)$
d) $O(\log n) < O(n) < O(n*\log n) < O(n^2) < O(2^n)$

135. In quick sort, a best /desirable choice of pivot for partitioning the list will be
a) First element of the list b) Last element of the list
c) Median of the list c) A randomly chosen element of the list

136. The time complexity of binary search is
a) $O(n^2)$ b) O(n) c) O(log n) d) O(n log n)

137. The fastest sorting algorithm for an almost already sorted array is
a) quick sort b) merge sort c) selection sort d)insertion sort

138. For merging two sorted listed of sizes m and n into a sorted list of size m+n, we require comparison of
a) O(m) b) O(n) c) O(m+n) d) O(log(m)+log(n))

139. The running time of an algorithm T(n),where n is the input size is given by
$T(n) = 8 T(n/2) + qn$, if $n>1$
$= p$, if $n=1$
where p and q are constants. The order of this algorithm is
a) n^2 b) n^n c) n^3 d) n

Module – VII/Hashing

140. Ratio of number of items in hash table , to the table size is called
a) Load factor b) Item factor c) Balanced factor d) All of these

141. Which of the following methods has the best average case complexity for searching?
a) Hashing b) Sequential c) Random d)Binary

142. The technique of linear probing for Collision Resolution can lead to

a) clustering b) Efficient storage utilization c) overflow
d) underflow

143. Which of the following is not a requirement of good hashing function?
a) Avoid collision b) Reduce the storage space
c) Make faster retrieval d) None of these

144. Which of the following is not related to hashing?
a) Synonyms b) Collision c)Balance factor d)Load factor

145. The Linear Probing Technique for collision resolution can lead to
a) Primary clustering b) Secondary clustering
c) Overflow d) Efficiency storage utilization

146. The average search time of hashing with linear probing will be less if the load factor :
a) is far less than one
b) equals one
c) is far greater than one
d) none of these

147. A hash table can store a maximum of 10 records. Currently there are records in locations 1, 3, 4, 7, 8, 9, 10.The probability of a new record going into location 2, with a hash function resolving collisions by linear probing is
a) 0.6 b) 0.1 c) 0.2 d) 0.5

148. Consider a hash function that resolves collision by quadratic probing. Assume the address space is indexed from 1 to 8.Which of the following locations will never be probed if a collision occurs at position 4?
a) 2 b) 4 c) 5 d) 8

149. A hash table has space for 100 records. What is the probability of collision before the table is 100 % full?
a) 0.45 b) 0.5 c) 0.3 d) 0.34 (Approximately)

150. A hash function f defined as f (key) = key mod 7, with linear probing, is used to insert the keys 37, 38, 72, 48, 98, 11, 56 into a table indexed from 0 to 6.What will be location of key 11?
a) 1 b) 4 c) 5 d) none of these

Part B

Short Answer Type Questions [5 Marks]

Array

1. What is Abstract Data type? What do you mean by Dynamic Data structure?
2. What are sparse matrices? How is it represented in memory? What are the types of sparse matrices?
3. Let the size of the elements stored in an 8 x 3 matrix be 4 bytes each. If the base address of the matrix is 3500, then find the address of A [5, 2] for both row major & column major cases.

Stack and Queue

4. What is stack? Why it is called LIFO? What is top of the stack? What are the conditions for stack underflow & stack overflow? Why stack is called ADT?
5. Write an algorithm to insert an element into a stack. Write an algorithm to delete an item from a stack.
6. What are the applications of stacks? What is linked stack?
7. What is queue? Why queue is called FIFO? What is linked queue?
8. Write the insert & delete functions for the queue.

9. What is circular queue? Write Q-insert algorithm for circular queue.
10. What is circular queue? Why we need circular queue? In which data structure insertion & deletion of elements can take place from either end? Explain.
11. **Write short notes on Priority Queue.

Linked List

12. Point out the difference between array & linked list.
13. Represent the following polynomials by a linked data structure (show only the diagram) $- 5^{x5} + 4^{x4} - 25^{x3} + 10$.

14. What is linked stack & what is linked queue? Explain insertion & deletion operations on linked stack & linked queue.

Recursion

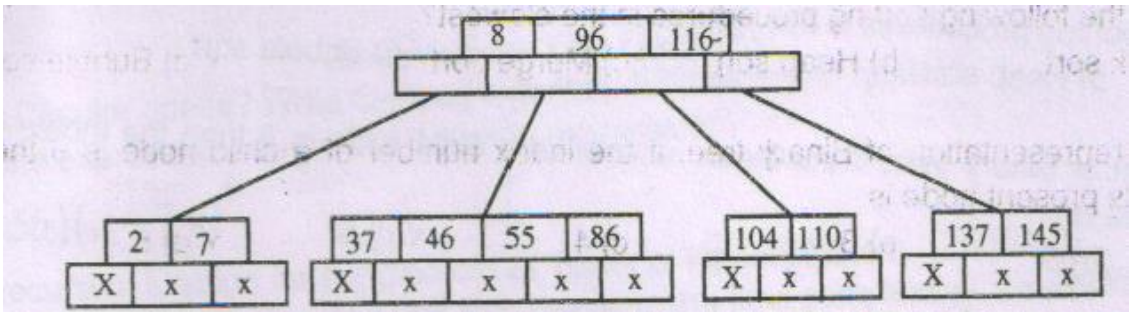
15. What is recursion tree? Write down the recursive definition for generation of the Fibonacci sequence. Assuming Fib (n) as a recursive function draw a recursive tree for Fib (6).
16. Write short notes on Tail Recursion & Tower of Hanoi Problem.
17. Write the difference between recursion & iteration. State the advantages and disadvantages of both the types.
18. What is recursion? Explain with an example. Explain: - "Recursion is worse than Iteration".
19. Are recursive routines more efficient than non-recursive routines? Justify your answer with suitable example.
20. Write the recursive function for the problem of Tower of Hanoi problem.

Tree

21. Define the following: (Each definition carries 1 mark)

Tree	*Node*	*Root*	*Degree of a Tree*	*Degree of a Node*	*Level*	*Directed and Undirected Graph*
Path	*Forest*	*Height*	*Ancestors*	*Descendant*	*Siblings*	*Weighted and Un-weighted Graph*
Terminal Nodes	*Non-Terminal Nodes*	*Internal Path*	*External Path*	*Leaf Node*	*Orchard*	*Sub-graph*
Degree of a Graph	*Cut Vertex*	*Articulation Point*	*Pendant Vertex*	*Clique*	*Complete and Connected Graph*	*Path and Isomorphism*

22. For a non-empty binary tree prove $n_0=n_2+1$, where n_0= No. of terminal nodes & n_2= No. of nodes of degree 2.
23. The maximum number of nodes on level "i" of a binary tree is ^{2i} (i>0).
24. What is B-Tree? Explain with an example. Write the advantages & disadvantages of B-Tree. What are the uses of such a tree in data structures?
25. What is threaded binary tree? Write the memory representations of threaded binary tree.
26. Write an algorithm to delete a node from binary search tree.
27. What is BST?
28. Consider a B-Tree of order 5 as shown below- insert the elements 4, 5, 58, 6 in this order in the B-Tree.



(5)

29. Write a c language function to find the in-order successor of the root of a binary tree.

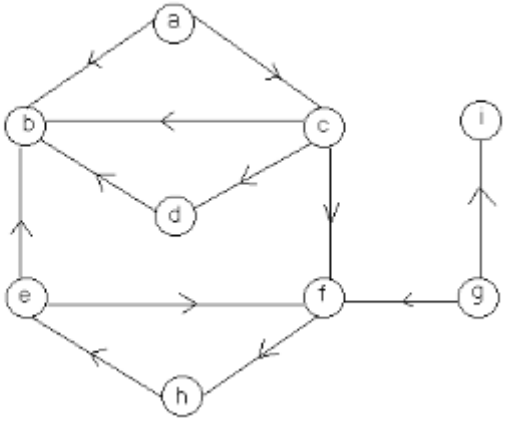
30. Write short notes on the following:
    - AVL Tree.
    - B+ Tree.
    - B* Tree.
    - BST.
    - Threaded Binary Tree.
31. Write the non-recursive functions for in-order traversal of binary tree.
32. Show the stages in growth of an order- 4 B-Tree when the following keys are inserted in the order given:
    74, 72, 19, 87, 51, 10, 35, 18, 39, 60, 76, 58, 19, 45
33. Write the non-recursive functions for pre-order traversal of binary tree.
34. Write the non-recursive functions for post-order traversal of binary tree.
35. How do AVL trees differ from binary search tree?
    Insert the following keys in the order given below to build them into an AVL tree.
    8, 12, 9, 11, 7, 6
    Clearly mention the different rotations used and balance factor of each node.
36. Construct the expression tree for the following expression tree : $E = (2a + 5b)(x - 7y)^4$.

Searching, Sorting and Time –Space Complexity

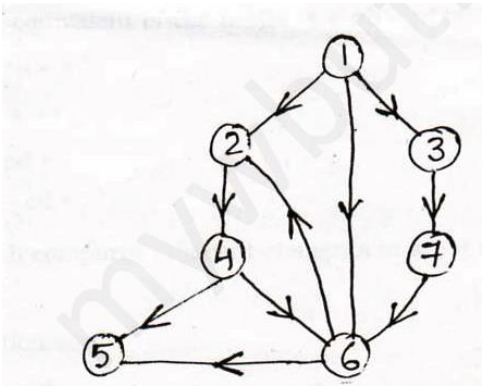
37. What is the prerequisite for binary search? What are the advantages of binary search over linear search?
38. "Binary search technique can't be implemented using linked list" .Justify.
    Derive the worst case time complexity of Binary search.
39. What do you mean by internal & external sorting? Explain with an example.
40. What is heap? Define max and min heap. Explain with an example how to construct a heap (show both types).
41. Write the algorithm to sort an array of integers using **Insertion Sort** method. Explain the **time complexity** of this sorting algorithm.
42. Explain with a suitable example, the principle operation of **Quick sort**.
43. Write a c function for **selection sort** and also calculate the **time complexity** for selection sort.
44. Explain with a suitable example the principles of operation of **Heap sort**. Find the **timecomplexity** of the algorithm.
45. Define Big-O, Ω, Θ notation.

Graph

46. What is a complete graph? Show that the sum of degree of all the vertices in a graph is always even.
47. Compare BFS and DFS. Discuss the two different ways of representing a graph.
48. Give the adjacency matrix and adjacency list of the following graph.



49. Prove that the maximum number of edges possible in a simple graph of n nodes is n(n-1)/2.
50. For the following graph Find
    i) BFS traversal
    ii) DFS traversal



Hashing

51. Explain with suitable example the collision resolution scheme using linear probing with open addressing.
52. What do you mean by hashing? What are the applications where you will prefer hash tables to other data structures? What do you mean by collision? How is it handled?
53. Why the hash functions need to be simple? What is the primary advantage of hashing over deterministic search algorithms?
54. Define hashing. Why hashing is referred to as heuristic search method?
55. Explain linear probing and double hashing with suitable example.
56. Write an algorithm to insert an element in the hash table using Quadratic probing.

Part C

Stack and Queue

1.

a) Consider the following stack of characters, where STACK is allocated N=8 memory cells: STACK: A, C, D, F, K, _, _, _. Describe the stack as the following operations take place: (a) POP(STACK, ITEM) (b) POP(STACK, ITEM) (c) PUSH(STACK,L) (d) PUSH(STACK,P) (e) POP(STACK, ITEM) (f) PUSH(STACK,R) (g) PUSH(STACK,S) (h) POP(STACK, ITEM).
[T:5,M:4]

b) Consider the following queue of characters, where queue Q is a circular array which is allocated 5 memory cells: Front = 2, Rear = 3, Q: _, P, Q, _, _. Describe the following operations on queue: - (a) R is added to the queue. (b) Two letters are from the queue. (c) S, T, U are added to queue. [**T:5,M:3**]

2. Write logic for reverse the order of elements on a stack S **[T;5,M:5]**

3. Suppose that you have a stack and push to the stack the integers 1,2,...n in that sequence. In between these push operations you also invoke some pop operations in such a way that pop request is never set to an empty stack. Immediately before each pop operation you also print the top of the stack. After all of the integers 1, 2 ...n are pushed, the elements remaining in the stack are printed and popped resulting in an eventually empty stack. If n=5 it means1, 2,3,4,5 then what is the printed sequence? **[T:5,M:5]**

4.
   a) Convert A+ (B*C-(D/E^F)*G)*H into its **postfix** form. Evaluate: 562+*124/- using stack. **[T:10,M:8]**
   b) The following postfix expression with single digit operands is evaluated using stack
      8 2 3 ^ / 2 3 * + 5 1 * -
      Note that ^ is the exponentiation operator.
      What will be the top two elements of the stack after the first * is evaluated.
      Convert A $ B * C- D +E / F/(G+G) into **prefix**. **[T:10, M:8]**

Linked List

5. The following C function takes a singly linked list of integers as a parameter and rearranges the elements of the list. The function is called with the list containing the integers 1,2,3,4,5,6,7 in the given order. What will be the contents of the list after the function completes execution? **[T:4,M:5]**

```
struct node
{
        int value;
        struct node *next;
};
void rearrange(struct node *list)
{
        struct node *p,*q;
        int temp;
        if(!list||!list->next)
                return;
        p=list;
        q=list->next;
        while(q)
        {
                temp=p->value;
                p->value=q->value;
                q->value=temp;
                p=q->next;
                q=p?p->next:0;
        }
}
```

6. Why ***Circular linked list*** is important in case of **Josephus** Problem?
7. Supposed the linked list in the memory consisting of numerical values. Write a function for each of the following: **[T:15,M:12]**
   i) To find the maximum (MAX) of the values in the list.
   ii) To find the average (MEAN) of the values in the list.

iii) To find the product (PROD) of the values in the list.

Recursion and Tree

8. Consider the following C program segment **[T:5,M:5]**

```
structCellNode
{
        structCellNode *leftchild;
        int element;
        structCellNode *rightchild;
};
intDoSomething(structCellNode *ptr)
{
        int value=0;
        if(ptr!=NULL)
        {
                if(ptr->leftchild!=NULL)
                        value=1+DoSomething(ptr->leftchild);
                if(ptr->rightchild!=NULL)
                        value=max(value,1+DoSomething(ptr->rightchild));
        }
        return value;
}
```

What will be the value returned by the function Do Something when a pointer to the root of a non-empty tree is passed as the argument.

9. Consider the following C program segment where Cell Node represents a node in a binary tree **[T:5,M:5]**

```
structCellNode
{
                structCellNode *leftchild;
                int element;
                structCellNode *rightchild;
};
intGetValue(structCellNode *ptr)
{
        int value=0;
        if(ptr1=NULL)
        {
                if((ptr->leftchild==NULL)&&(ptr->rightchild==NULL))
                        value=1;
                else
                        value=value+GetValue(ptr->leftchild)+GetValue(ptr->rightchild);
        }
        return value;
}
```

What will be the value returned by GetValue when a pointer to the root of a binary tree is passed as its argument.

10. Show the steps in creation of a height balanced binary AVL tree using insertion of items in the following order-show the steps required with diagrams..

(March, May , November, August, April, January, December, July, February, june, October, September) **[T: 12, M: 10]**

11. Construct a B-tree of **order 3** with the following data **[T:5,M:5]**
    50, 40, 60, 30, 70, 20, 80, 10, 90, 9, 99
    Insert the following keys into a B-Tree of given order mentioned below: - **[T:5,M:5]**
    a, f, b, k, h, m, e, s, r, c. (**Order 3**)
    a, g, f, b, k, d, h, m, j, e, s , i, r, x, c, l, n, t, u, p. (**Order 5**)
    [T:10,M:8]
12. Show how the letters A to P of English alphabet can be entered into a b-Tree of **order 4**.

13. Construct a binary tree from pre and inorder traversal**.** **[T:5,M:5]**
    - Pre-order- A B D I E J C F G K
    - In-order - D I B E J A F C K G

14. Construct a Binary Tree from Inorder and Postorder traversal. **[T:5,M:5]**
    - Post-order- B C A P N T L K G F D
    - In-order - A B C D F G K L N P T

15. The degree of a node is the number of children it has. Show that in any binary tree, the numbers of leaves are one more than the number of nodes of degree 2. **[T:10,M:8]**

16. Show how the following integers can be inserted in an empty binary search tree in the order they are given: 50, 30, 10, 90, 100, 40, 60, 20, 110, 5
    [T:2,M:3]
17. What are the problems of binary tree? Explain how a height-balanced tree can be formed by inserting the following elements in the given order:
    1, 2, 3, 4, 5, 6, 8, 9, 10, 7, 11
    Show the **root** element that can be **deleted** from the above tree. **[T:7,M:8]**

18.
    a) Draw the expression tree of the following infix expression. Convert it into prefix and postfix expressions.
    ((a+b) +c*(d+e) +f)*(g+h) **[T: 7, M: 7]**
    b) Consider the algebraic expression
    $E=(5x+z)(3a-b)^2$
        i) Draw the expression tree corresponding to E
        ii) Find the scope of exponential operator i.e. the sub tree rooted at the exponential operator. **[T:5,M:5]**

    a) Which one of the following is a valid sequence of elements in an array representing 3 –ary max heap?
        i) 1,3,5,6,8,9
        ii) 9,6,3,1,8,5
        iii) 9,3,6,8,5,1
        iv) 9,5,6,8,3,1
    Explain why one option is correct and others fail. **[T: 5,M:5]**

Searching, Sorting and Time –Space Complexity

19. Show that the function f(n) defined by: -
    f (1) = 1
    f (n) = f (n-1) + 1/n for n>1, has the complexity O (log n) **[T:5,M:5]**

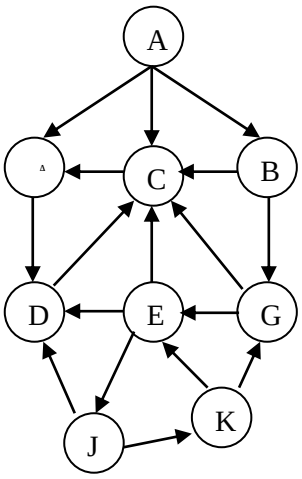
20. Show how the **merge sort** algorithm will sort the following array in increasing order**:**
    100, 90, 80, 70, 60, 50, 40, 30, 20 **[T:6,M:6]**
    Derive the **time complexity** of **merge sort** algorithm.

21. Explain with suitable example, the principal operation of **Quick sort.** **[T;15,M:15]**
Find the **Best case** and **worst case** complexity of **quick sort** algorithm.

22. Derive values related to average case and worst case behaviour of bubble Sort algorithm. Also confirm that the best case behaviour is O(n). **[T:8,M:8]**
23. Sort the following list using **Heap Sort**
66, 33, 40, 20, 50, 88, 60, 11, 77, 30, 45, 65
[T: 10, M: 8]

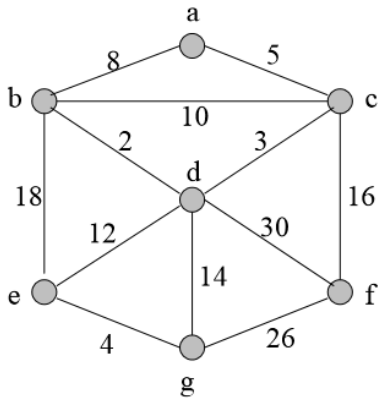
Graph

24. Apply BFS/DFS Algorithm the find out the path of the given graph: **[T:10,M:8]**

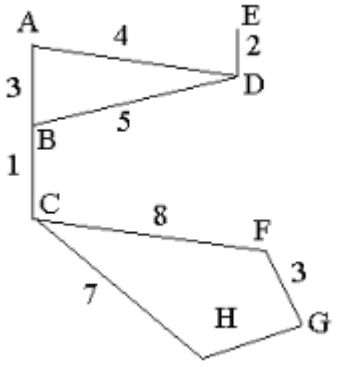


25.
a) Draw the minimum cost spanning tree for the graph given below and also find its cost.

[T: 6, M: 6]



b) Execute both Prim's and Kruskal's algorithm on the following graph to find the minimum cost.
[T:10,M:8]



0

Hashing

26.

a) A hash table of length 10 uses open addressing with hash function h(k)=k mod 10, and linear probing. After inserting 6 values into an empty hash table, the table is as shown below.

[T: 5, **M: 5]**

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

i) Which one of the following choices gives a possible order in which the key values could have been inserted in the table? Explain.

a) 46, 42, 34, 52, 23, 33
b) 34, 42, 23, 52, 33, 46
c) 46, 34, 42, 23, 52, 33
d) 42, 46, 33, 23, 34, 52

ii) How many different insertion sequences of the key values using the same hash function and linear probing will result in the hash table shown above?

b) The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function h(k)=k mod 10 and linear probing. What is the resultant hash table? **[T: 2, M: 2]**

1. An implementation of queue Q, using stacks S1 and S2 is given below: **[T:5,M:2]**

```
void insert (Q, x) {
      push (S1, x);
}
void delete (Q)
 {
      if (stack-empty (S2)) then
            if (stack-empty (S1)) then
             {
                  print ( " Q is empty" );
                  return;
             }
             else
                  while (! (stack-empty (S1))) then
```

```
        {
              x = pop (S1);
              push (S2, x);
        }
        x = pop (S2);
  }
```

Let *n* insert and *m (≤ n)* delete operations be performed in an arbitrary order on an empty queue Q. let *x* and *y* be the number of push and pop operations performed respectively in the process. Which one of the following is true for all *m* and *n*?

(A) $n + m \leq x < 2n$ and $2m \leq y \leq n + m$ (B) $n + m \leq x < 2n$ and $2m \leq y \leq 2n$
(B) $2m \leq x < 2n$ and $2m \leq y \leq n + m$ (D) $2m \leq x < 2n$ and $2m \leq y \leq 2n$

2. Assume that the operators +, -, × are left associative and ^ is right associative. The order of precedence (from highest to lowest) is **^, ×, +, -.**What will be the postfix expression corresponding to the infix expression a+b ×c-d ^ e ^f ? **[T:5,M:2]**
3. Explain the execution order of factorial (5) using stack**.** **[T:2,M:2]**

4. Pick out the equivalent postfix expression for this infix expression:
   X=((A+B)*C-(D-E)^(F+G)) **[T:5,M:3]**
5. Assume that a queue is available for pushing and popping elements. Given an input sequence a, b, c, (c be the first element), give the output sequence of elements if the rightmost element given above is the first to be popped from the queue. **[T:2,M:2]**

Linked List

6. Is linked list can be considered as a linear data structure? Explain. **[T:2,M:2]**
7. If you are using C language to implement the heterogeneous linked list, what pointer type will you use?**[T:2,M:2]**
8. What is self –referential structure? Explain with an example. **[T:2,M:2]**
9. Write a function to remove the first node from the list and insert it at end, without changing the info part of any node.**[T:7,M:3]**
10. Which of the following statements are true in the case of doubly linked list?
    i) Every node is connected to other node
    ii) We can traverse in both the directions
    explain with example. **[T:5, M:2]**

11. What does the following function do for a given Linked List with first node as *head*?
    [T:2,M:2]

```
voidfun1(structnode* head)
{
  if(head == NULL)
    return;
  fun1(head->next);
  printf("%d  ", head->data);
}
```

12. Can we perform insertion sort in linked lists? Explain. **[T:4, M:3]**
13. The following function reverse () is supposed to **reverse** a singly linked list. There is one line missing at the end of the functio**n.**
    [T:4,M:3]

```
/* Link list node */
structnode{
    intdata;
    structnode* next;
};

/* head_ref is a double pointer which points to head (or start)  pointer
of linked list */
staticvoidreverse(structnode** head_ref)
{
    structnode* prev   = NULL;
    structnode* current = *head_ref;
    structnode* next;
    while(current != NULL)
    {
        next  = current->next;
        current->next = prev;
        prev = current;
        current = next;
    }
    /*ADD A STATEMENT HERE*/
}
```

What should be added in place of "/***ADD A STATEMENT HERE***/", so that the function correctly reverses a linked list.

14. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is which of the following? Then explain**.** **[T:1,M:2]**

    (A) log 2 n
    (B) n/2
    (C) log 2 n – 1
    (D) n

15. Consider the function f defined below.

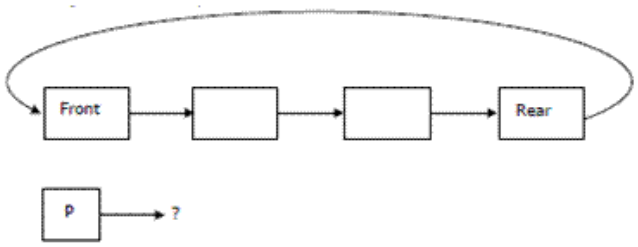
    [T:2,M:2]

```
struct item
{
  int data;
  struct item * next;
};

int f(struct item *p)
{
  return (
          (p == NULL) ||
          (p->next == NULL) ||
          (( P->data <= p->next->data) && f(p->next))
         );
}
```

    For a given linked list p, when the function f returns 1 ?

16. A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. To which node should p point such that both the operations enQueue and deQueue can be performed in constant time? **[T:2,M:2]**

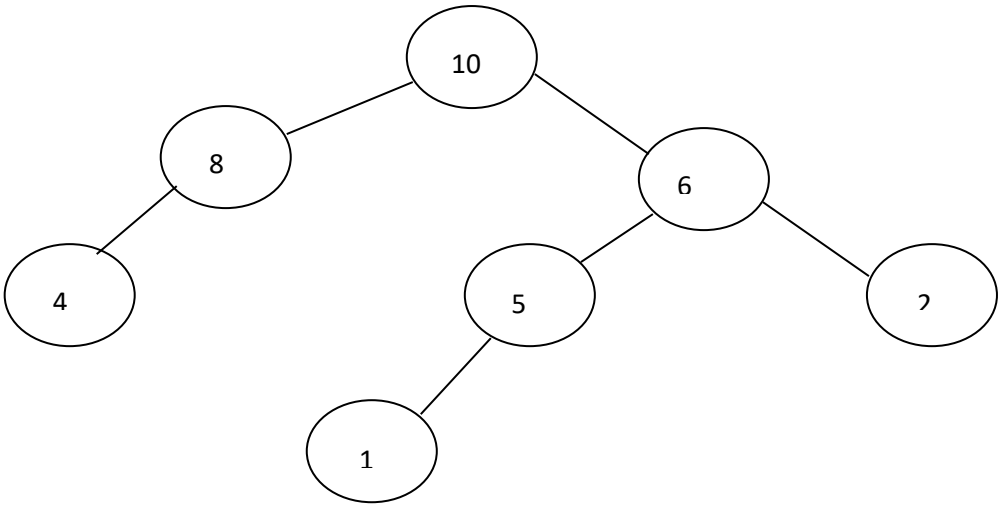


17. Is it possible to find a loop in a Linked list? Give reason to your answer. **[T:3,M:3]**
18. How would you sort the elements in a single linked list?**[T:6,M:3]**
19. Suppose a given linked-list has some odd numbers and even numbers. Write an algorithm to find them and position of them. **[T:7,M:3]**
20. Which operations are more effective in linked list than arrays? And why?
    i) Insertion
    ii) Deletion
    iii) Traversal **[T:2,M:2]**
21. Can we do a Binary search on a linked list? Give reason to your answer. **[T:5,M:2]**
22. How to read a singly linked list in backward? **[T:7,M:3]**
23. If you implement stack or queue using linked list, then what is the full conditions? **[T:5,M:3]**
24. Is Circular linked list is infinite? Explain. **[T:5,M:3]**
25. Application of single and double linked list. **[T:2,M:3]**
26. Who is responsible to dynamically allocate the memory for creation of a node in a single linked list? **[T:2,M:2]**

Tree

27. What is complete binary tree and what is full binary tree? **[T:3,M:3]**
28. Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty binary search tree. The binary search tree uses the usual ordering on natural numbers. What is the in-order traversal sequence of the resultant tree?**[T:3 M:2]**
29. What is the maximum possible number of nodes in a binary tree at level 6? **[T:2,M:2]**
30. The preorder traversal sequence of a binary search tree is 30, 20, 10, 15, 25, 23, 39, 35, 42. What is the post order traversal sequence of the same tree? **[T:4,M:3]**

31. Check the following is a max-heap or not? Explain. **[T:2,M:2]**



32. Which one of the following array represents a binary max-heap? **[T:2,M:2]**

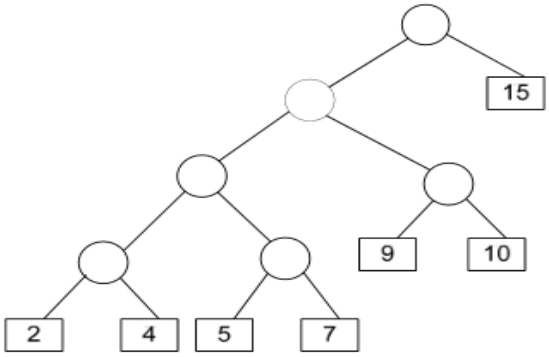
a) {25,12,16,13,10,8,14}
b) {25,14,13,16,10,8,12}
c) {25,14,16,13,10,8,12}
d) {25,14,12,13,10,8,16}

33. What is the content of the array after two delete operations on the correct answer to the previous question? **[T:2,M:2]**
    a) {14,13,12,10,8}
    b) {14,12,13,8,10}
    c) {14,13,8,12,10}
    d) {14,13,12,8,10}

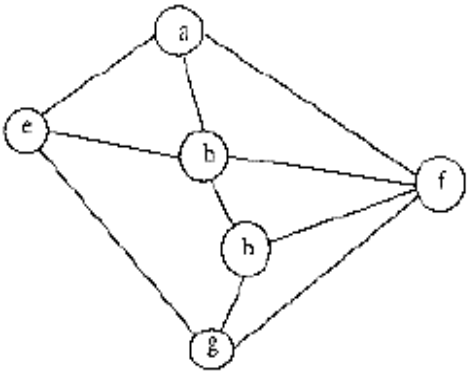
34. Consider the following nested representation of binary trees: (X Y Z) indicates Y and Z are the left and right sub stress, respectively, of node X. Note that Y and Z may be NULL, or further nested. Which of the following represents a valid binary tree? **[T:10,M:4]**
    (a) (1 2 (4 5 6 7))
    (b) (1 (2 3 4) 5 6) 7)
    (c) (1 (2 3 4)(5 6 7))
    (d) (1 (2 3 NULL) (4 5))
35. The following numbers are inserted into an empty binary search tree in the given order: 10, 1, 3, 5, 15, 12, and 16. What is the height of the binary search tree (the height is the maximum distance of a leaf node from the root)? **[T:3, M:4]**
36. What is the number of leaf nodes in a rooted tree of n nodes, with each node having 0 or 3 children? **[T:2,M:2]**
37. Why tree is called non-linear data structure? **[T:2, M:2]**
38. Why heap tree is represented with array? **[T:2, M:3]**
39. Write some applications of tree. **[T:2, M:2]**
40. The height of a binary tree is the maximum number of edges in any root to leaf path. What is the maximum number of nodes in a binary tree of height h? **[T:2,M:2]**
41. In a binary tree with n nodes, every node has an odd number of descendants. Every node is considered to be its own descendant. What is the number of nodes in the tree that have exactly one child? **[T:2,M:2]**
42. Consider a node X in a Binary Tree. Given that X has two children, let Y be In order successor of X.Can Y have any left child? **[T:2,M2]**
43. Post order traversal of a given binary search tree, T produces the following sequence of keys 10, 9, 23, 22, 27, 25, 15, 50, 95, 60, 40, 29 What will be the result of an in-order traversal of the tree T? **[T:2,M2]**

44. A scheme for storing binary trees in an array X is as follows. Indexing of X starts at 1 instead of 0.The root is stored at X [1].For a node stored at X[i], the left child, if any, is stored in X [2i] and the right child, if any, in X[2i+1]. To be able to store any binary tree on n vertices what is the minimum size of X should be. **[T:2,M2]**

45. A complete n-ary tree is a tree in which each node has n children or no children. Let I be the number of internal nodes and L be the number of leaves in a complete n-ary tree. If L = 41, and I = 10, what is the value of n? **[T:7,M:4]**
46. What is the weighted external path length of the binary tree shown in the following figure? **[T:2, M:2]**



47. Consider the following graph **[T:2,M:2]**



Among the following sequences
I) a b e g h f
II) a b f e h g
III) a b f h g e
IV) a f g h b e
Which are depth first traversals of the above graph?

Searching and Sorting

48. What is the difference between Linear & Binary Search? What is the prerequisite for binary search? **[T:3,M:3]**
49. "Binary search technique can't be implemented using linked list" .Justify **[T:2,M:2]**
50. What do you mean by complexity? What is time complexity and what is space complexity? **[T:3,M:3]**
51. Explain the meaning of worst case analysis and best case analysis with an example**.** **[T:2,M:2]**
52. Which sorting algorithm is easily adaptable to singly linked lists? Justify. **[T:3,M:3]**
53. You have an array of n elements. Suppose you implement quick sort by always choosing the central element of the array as the pivot. Then what is the tightest upper bound for the worst case performance? **[T:2,M:2]**
54. In which situation insertion sort running time complexity O(n). Explain with an example. Then same situation for quick sort what happen explain. **[T:10,M:4]**
55.
56. Let P be a Quick Sort Program to sort numbers in ascending order using the first element as pivot. Let t1 and t2 be the number of comparisons made by P for the inputs {1, 2, 3, 4, 5} and {4, 1, 5, 3, 2} respectively. Which one of the following holds? **[T:2,M:2]**
    (A) t1 = 5

(B) t1 < t2
(C) t1 > t2
(D) t1 = t2

57. Bubble sort and selection sort has the worst case time complexity $O(n^2)$ but selection sort is faster than bubble sort. Why? **[T:2,M:2]**
58. Why merge sort is called divide and conquer method?**[T:2,M:2]**
59. How many key comparisons and assignments an insertion sort makes in its worst case? **[T:2,M:2]**

60. Sort the following sequence of keys using merge sort.
    66, 77, 11, 88, 99, 22, 33, 44, 55 [**T:5,M:3]**
61. Which sorting algorithm is best if the list is already sorted? Why? **[T:2,M:2]**
62. What is the time complexity (in all cases) of Merge sort and Heap sort algorithms? **[T:2,M:3]**

Hashing

63. Given the following input (4322, 1334, 1471, 9679, 1989, 6171, 6173, and 4199) and the hash function x mod 10, which of the following statements are true? **[T:1,M:2]**

    i. 9679, 1989, 4199 hash to the same value
    ii. 1471, 6171 has to the same value
    iii. All elements hash to the same value
    iv. Each element hashes to a different value

64. What do you mean by hashing? What are the applications where you will prefer hash tables to other data structures? **[T:2,M:2]**
65. Write down some applications of hashing. **[T:2,M:2]**
66. Is hashing a search technique? If yes, then compare with binary search. **[T:5,M:3]**
67. What are different methods of collision resolution in hashing? **[T:2,M:2]**
68. What are the advantages and disadvantages of hashing with respect to searching? **[T:2,M:2]**
69. What do you mean by probing? **[T:2,M:2]**
70. What is the significance "open" in open addressing technique? **[T:2,M:2]**
71. Why complexity of hashing is O(1)? **[T:2,M:2]**