**Unit-3**

**Linked Lists**

1. The operation of processing each element in the list is known as

a. Sorting b. Merging c. Inserting d. Traversal

1. 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
2. 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

1. 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 b) Pointer to integer c) Pointer to node d) Node
2. 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(n2) d) O(n3)
3. 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)
4. What would be the asymptotic time complexity to find an element in the linked list?  
   a) O(1) b) O(n) c) O(n2) d) O(n4)
5. 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)
6. 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

1. 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

1. Linked lists are not suitable for the implementation of \_\_\_\_\_\_\_\_\_\_\_  
   a) Insertion sort b) Radix sort

c) Polynomial manipulation d) Binary search

1. Linked list is considered as an example of \_\_\_\_\_\_\_\_\_\_\_ type of memory allocation.  
   a) Dynamic b) Static c) Compile time d) Heap
2. In Linked List implementation, a node carries information regarding \_\_\_\_\_\_\_\_\_\_\_  
   a) Data b) Link c) Data and Link d) Node
3. Linked list data structure offers considerable saving in \_\_\_\_\_\_\_\_\_\_\_\_\_  
   a) Computational Time b) Space Utilization  
   c) Space Utilization and Computational Time d) Speed Utilization
4. 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
5. 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
6. n 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/2c) log 2 n – 1 d) n
7. 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
8. 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

1. 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
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)
3. 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

23.