Linked Lists in Python

Q1 Let P be a singly linked list. Let Q be the pointer to an intermediate node x in the list. What is the worst-case time complexity of the best known algorithm to delete the node x from the list?

(A) O(n)
(B) $O((\log^2)n)$
(C) O(log n)
(D) O(1)

Q2 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 n
(B) n/2
(C) log n – 2
(D) n

Q3 What is the worst case time complexity of inserting n elements into an empty linked list, if the linked list needs to be maintained in sorted order ?

(A) Θ(n)
(B) Θ(n log n)
(C) $\Theta(n^2)$
(D) Θ(1)

Q4 In a circular linked list oraganisation, insertion of a record involves modification of

(A) One pointer.
(B) Two pointers.
(C) Multiple pointers.
(D) No pointer.

Q5 Which of the following operations is performed more efficiently by doubly linked list than by singly linked list?

(A) Deleting a node whose location in given
(B) Searching of an unsorted list for a given item
(C) Inverting a node after the node with given location
(D) Traversing a list to process each node

Q6 Consider an implementation of unsorted singly linked list. Suppose it has its representation with a head at first node and tail reference at last node.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

Q7 Consider an implementation of unsorted singly linked list. Suppose it has its representation with a head pointer only. 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

Q8 What does the following function do for a given Linked List with first node as head?

```
class Node:
def __init__(self, data):
        self.data = data
        self.next = None
def fun1(head):
    if head.next is None:
        return
    print(head.next.data, end='  ')
    fun1(head.next)
```

(A) Prints all nodes of linked lists
(B) Prints all nodes of linked list except first node
(C) Prints alternate nodes of Linked List

(D) Prints all nodes of linked list except last node

Q9 What does the following function do for a given Linked List with first node as head?

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
def fun1(head):
    if head is None:
        return
    print(head.data, end='  ')
    fun1(head.next)
```

(A) Prints all nodes of linked lists

(B) Prints all nodes of linked list except first node

(C) Prints alternate nodes of Linked List

(D) Prints all nodes of linked list except last node

Q10 What does the following function do for a given Linked List with first node as head?

```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
def fun1(head):
    if head is None:
        return
    fun1(head.next)
    print(head.data, end='  ')
```

(A) Prints all nodes of linked lists

(B) Prints all nodes of linked list in reverse order

(C) Prints alternate nodes of Linked List

(D) Prints all nodes of linked list except last node

Answer Key

Q1	A
Q2	D
Q3	C
Q4	B
Q5	A
Q6	C
Q7	B
Q8	B
Q9	A
Q10	B

Hints & Solutions

Note: scan the QR code to watch video solution

Q1 Text Solution:
A

Q2 Text Solution:
D

Q3 Text Solution:
C

Q4 Text Solution:
B

Q5 Text Solution:
A

Q6 Text Solution:
C

Q7 Text Solution:
B

Q8 Text Solution:
B

Q9 Text Solution:
A

Q10 Text Solution:
B