

### Quiz On LinkedList with Answers with Explanation wherever possible

Q1) Consider a singly linked list of the form where F is a pointer to the first element in the linked list and L is the pointer to the last element in the list. The time of which of the following operations depends on the length of the list?

- a) Delete the element from the last
- b) Delete the element from first
- c) Add an element to the last
- d) Interchange the first two elements

Answer is a)

Explanation: we need to traverse the list completely to get the second last node to remove the last node.

Q2) In what best complexity you can find the kth element from last in a linked list?

- a) Time:  $O(n^2)$  Space:  $O(1)$
- b) Time:  $O(n)$  Space:  $O(n)$
- c) Time:  $O(n)$  Space:  $O(1)$
- d) None

Answer is c)

Explanation: you can see the video of remove Kth element from last.

Q3) Which of the following points is/are true about Linked List data structure when it is compared with 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) The size of array has to be pre-decided, linked lists can change their size any time
- e) All of above

Answer is e) all the above

Q4) 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: Singly linked list cannot be answer because we cannot find last element of a singly linked list in  $O(1)$  time.

Doubly linked list cannot also not be answer because of the same reason as singly linked list.

Q5) Which of the following sorting algorithms can be used to sort a random linked list with minimum time complexity?

- a) insertion sort
- b) merge sort
- c) quick sort
- d) heap sort

Answer is b) merge sort

Q6) Predict the output that the following code gives for a linked list given as 1->2->3->4->5->6

```
void fun(node start)
{
    if(start == null)
        return;
    System.out.print(start.data+" ");
    if(start.next != null )
        fun(start.next.next);
    System.out.print(start.data+" ");
}
```

- a) 1 4 6 6 4 1
- b) 1 3 5 1 3 5
- c) 1 2 3 5
- d) 1 3 5 5 3 1

Answer is d) using recursion

Q7) Skip lists are similar to which of the following data structure?

- a) Stack
- b) Heap
- c) BST
- d) Balanced BST

Answer is d) Balanced BST

Explanation: A skip list is a data structure that is used for storing a sorted list of items with a help of hierarchy of linked lists that connect increasingly sparse subsequences of the items. A skip list allows the process of item look up in efficient manner

Q8) What is a memory efficient double linked list?

- a) Each node has only one pointer to traverse the list back and forth
- b) An auxiliary singly linked list acts as a helper list to traverse through the doubly linked list
- c) None

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

Q9) Predict the modified doubly linked list of the following code for the given doubly linked list as 1<--> 2 <--> 3 <--> 4 <--> 5 <-->6

```
void fun(node head_ref)
{
    node temp = null;
    node current = head_ref;

    while (current != null)
    {
        temp = current.prev;
        current.prev = current.next;
        current.next = temp;
        current = current.prev;
    }

    if(temp != null )
        head_ref = temp.prev;
}
```

- a) 2 <--> 1 <--> 4 <--> 3 <--> 6 <-->5
- b) 5 <--> 4 <--> 3 <--> 2 <--> 1 <-->6
- c) 6 <--> 5 <--> 4 <--> 3 <--> 2 <--> 1
- d) 6 <--> 5 <--> 4 <--> 3 <--> 1 <--> 2

Answer is option c) 6 <--> 5 <--> 4 <--> 3 <--> 2 <--> 1

Explanation: The given function reverses the given doubly linked list.