SANFOUNDRY
Global Education & Learning

Data Structure Questions and Answers – Singly Linked List Operations – 1

« Prev Next »

This set of Data Structure Interview Questions & Answers focuses on “Singly Linked List Operations – 1”.

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
View Answer

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.

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
View Answer

Answer: b
Explanation: We know the head node in the given linked list. Insertion and deletion of eleme 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).

3. 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
View Answer

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.

4. 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)
View Answer

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).

5. 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(n^3)
View Answer

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).

6. What would be the asymptotic time complexity to find an element in the linked list?
a) O(1)

b) O(n)
c) $O(n^2)$
d) $O(n^4)$

View Answer

Answer: b
Explanation: If the required element is in the last position, we need to traverse the entire linked list. This will take O (n) time to search the element.

7. 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) $O(n^2)$
d) $O(n^3)$

View Answer

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).

8. 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

View Answer

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.

9. Consider the following definition in c programming language

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

```
typedef struct node NODE;
NODE *ptr;
```

Which of the following c code is used to create new node?
a) ptr = (NODE*)malloc(sizeof(NODE));
b) ptr = (NODE*)malloc(NODE);
c) ptr = (NODE*)malloc(sizeof(NODE*));
d) ptr = (NODE)malloc(sizeof(NODE));

View Answer

Answer: a
Explanation: As it represents the right way to create a node.

Sanfoundry Global Education & Learning Series – Data Structure.

To practice all areas of Data Structure for Interviews, [here is complete set of 1000+ Multiple Choice Questions and Answers](#).

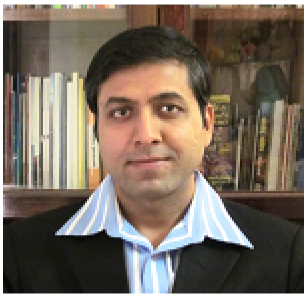
« Prev - Data Structure Questions and Answers – Queue Operations

» Next - Data Structure Questions and Answers – Singly Linked List Operations – 2

Recommended Posts:

1. Java Programming Examples on Collections
2. C Programming Examples on Trees
3. C# Programming Examples on Sorting
4. C Programming Examples on Bitwise Operations
5. Java Programming Examples on Combinatorial Problems & Algorithms
6. Data Science Questions and Answers
7. C++ Programming Examples on Combinatorial Problems & Algorithms
8. C Programming Examples on Stacks & Queues
9. C Tutorials
10. C Programming Examples using Recursion

11. C Programming Examples on Combinatorial Problems & Algorithms
12. Data Structures & Algorithms II – Questions and Answers
13. C Programming Examples without using Recursion
14. C# Programming Examples on Data Structures
15. Java Programming Examples on Data-Structures
16. C Programming Examples on Data-Structures
17. C++ Programming Examples on Data-Structures
18. Data Structure Questions and Answers
19. Python Programming Examples on Linked Lists
20. C Programming Examples on Linked List



Manish Bhojasia, a technology veteran with 20+ years @ Cisco & Wipro, is Founder and CTO at Sanfoundry. He is Linux Kernel Developer & SAN Architect and is passionate about competency developments in these areas. He lives in Bangalore and delivers focused training sessions to IT professionals in Linux Kernel, Linux Debugging, Linux Device Drivers, Linux Networking, Linux Storage, Advanced C Programming, SAN Storage Technologies, SCSI Internals & Storage Protocols such as iSCSI & Fiber Channel. Stay connected with him @ LinkedIn