**National University of Computer and Emerging Sciences**



**Laboratory Manual**

*for*

# Data Structures Lab

|  |  |
| --- | --- |
| Course Instructor | Mr. Uzair Naqvi |
| Lab Instructor | Mr. Durraiz Waseem |
| Section | BCS-3H |
| Date | Sept 19, 2024 |
| Semester | Fall 2024 |

# Department of Computer Science

FAST-NU, Lahore, Pakistan

###### **Overview**

Doubly Linked List is a variation of Linked list in which navigation is possible in both ways, either forward and backward easily as compared to Single Linked List. Following are the important terms to understand the concept of doubly linked list.

**Link** − Each node of a linked list can store a data called an element.

**Next −** Each node of a linked list contains a link to the next link called Next.

**Prev −** Each node of a linked list contains a link to the previous link called Prev.

**LinkedList** − A Linked List contains the connection link to the first link called First and to the last link called Last.

###### Representation:

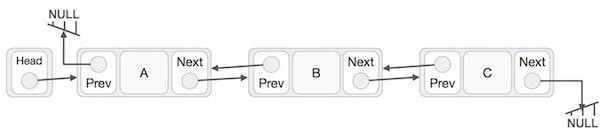


Figure 1. Doubly Linked List representation

A doubly linked list is a linked list in which every node has a next pointer and a back pointer. Every node contains the address of the next node (except the last node), and every node contains the address of the previous node (except the first node). A doubly linked list can be traversed in either direction.



**Part 1: Doubly Link List**

1. **Insert At Start** − Adds an element at the beginning of the list.
2. **Delete At Start** − Deletes an element at the beginning of the list.
3. **Insert At Last** − Adds an element at the end of the list.
4. **Delete At Last** − Deletes an element from the end of the list.
5. **Insert Before** − Adds an element before an item of the list.
6. **Delete** − Deletes an element from the list using the key.
7. **Display forward** − Displays the complete list in a forward manner.
8. **Display backward** − Displays the complete list in a backward manner.

###### Circular Linked List:

Circular linked list is a linked list where all nodes are connected to form a circle. There is no NULL at the end. A circular linked list can be a singly circular linked list or doubly circular linked list.

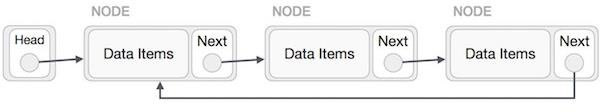


Figure 2. Singly List as Circular

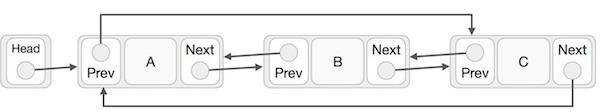


Figure 3. Doubly List as Circular

As per the above illustration, following are the important points to be considered.

* + The last link's next points to the first link of the list in both cases of singly as well as doubly linked list.
  + The first link's previous points to the last of the list in case of doubly linked list.

**Part 2: Circular linked List**

1. Function called **InsertAtStart** to add node at the beginning of Circular Linked list.
2. Function called **InsertAtEnd** to add node at the end of Circular Linked list.
3. Function called **deleteKey** to delete the node containing key in Circular Linked List.
4. Function called **deleteLast** to delete the last node in Circular Linked List
5. Function called **deleteAtIndex** to Delete the node from any given position
6. Function called **search** to search the key in Circular Link List. (Note: Take singly link list class for this task, containing only head pointer.)