

Linked Lists as a Data Structure

Lesson 4.3

Learning Objectives

- LO 4.3.1 **Utilize** linked list as an efficient storage medium of data
- LO 4.3.2 **Solve** computing problems effectively using linked lists

Linked Lists

- A linked list is a sequential-access data structure used for storing ordered elements
- They prioritize quick and easy data insertion and deletion over item lookup
- All linked lists are collections of node data structures that are connected by pointers — that's where the “link” in “linked list” comes from

Linked Lists

- Types of linked lists:

1. Singly linked lists – nodes are connected only through the “next” pointer and last node’s “next” pointer holds the NULL value.
2. Doubly linked lists – nodes are connected through the “next” and “prev” pointers, and the first node’s “prev” pointer and the last node’s “next” pointer holds the NULL value.
3. Circular linked lists – nodes are connected only through the “next” pointer, and the last node’s “next” pointer holds the address of the first node.

Singly vs. Doubly vs. Circular Linked Lists

	Singly	Doubly	Circular
Insert (Beginning)	$O(1)$	$O(1)$	$O(1)$
Insert (End)	$O(n)$	$O(1)$	$O(1)$
Insert (Random location)	$O(n)$	$O(n)$	$O(n)$
Delete (Beginning)	$O(1)$	$O(1)$	$O(1)$
Delete (End)	$O(n)$	$O(1)$	$O(1)$
Delete (Random location)	$O(n)$	$O(n)$	$O(n)$
Search	$O(n)$	$O(n)$	$O(n)$
Memory Space	$2n$	$3n$	$2n$

Linked Lists

- Doubly linked lists require more space per node and their basic operations are more expensive than for singly linked lists
- However, they are often easier to manipulate than singly linked lists because they allow for fast and easy sequential access to the list in both directions

Strengthening the Learning Objectives

LO 4.3.1 Utilize linked list as an efficient storage medium of data

LO 4.3.2 Solve computing problems effectively using linked lists

Implement the function

```
void addSorted(SINGLYLINKEDLIST sorted_list, int data)
```

where **sorted_list** is a sorted list of integers, and the function should add the **data** in the **sorted_list** wherein the **sorted_list**'s elements should still be sorted after the insertion.

LO 4.3.1 Utilize linked list as an efficient storage medium of data

LO 4.3.2 Solve computing problems effectively using linked lists

Create a program where a user has to repeatedly enter an integer. The program will halt if it encounters a -1 input from the user. The program then display all input values not including -1 in ascending order.

Example:

1 10 3 -4 2 6 1 90 2 -1

-4 1 1 2 2 3 6 10 90