Linked List

* Linear data structure
* Elements are not stored at contiguous memory locations
* Elements are linked using pointers
* Includes a series of connected nodes
* Each node stores the data and the address of the next node

Advantages over arrays:

* Array size is fixed; upper limit is fixed irrespective of usage
* Insertion/ Deletion of elements is difficult
* Implementation of stacks and queues
* Used for effective memory utilization, no memory wastage

Disadvantage over arrays:

* Random access is not allowed
* Extra memory space for a pointer is required with each element of the list
* Reverse traversing is not possible
* Searching an element takes O(n)
* Not cache friendly, since elements are not stored at contiguous memory locations.
* Takes a lot of time in traversing and changing the pointers
* Pointers are confusing to work on.

Representation:

A linked list is represented by a pointer (head) to the first node of the linked list.

Each node has 2 parts:

* A data item
* Pointer or reference to the next node (Address of next node)

In Java Linked List is represented as a class and a node as separate class.

In linked lists memory is allocated at run-time.

Applications:

Implementation of stacks and queues

Representation of trees and graphs

Representing sparse matrices

Manipulation of polynomials

Performing arithmetic operations on large integers

Finding paths in networks

Application of linked lists:

List of songs in music player

Web browser, previous and next web page urls are linked through previous and next buttons

Image viewer

Switching between tabs in windows using alt+tab

Contact list in mobile phones (inserted alphabetically)

Modifications in a document are created as nodes in doubly linked list makes ctrl+z possible

Advantages of Circular Linked List:

We can traverse from any node.

Implementation of queue.

When multiple applications have to run at a time.

Circular Doubly Linked Lists are used for implementation of advanced data structures like Fibonacci Heap.

Time Complexity Worst Case Average Case

* Search: O(n) O(n)
* Insert: O(1) O(1)
* Deletion: O(1) O(1)

Space Complexity

Search: O(n)

Insert: O(1)

Deletion: O(1)

Insertion:

Time Complexity: O(n)

Space Complexity: O(1)

Deletion:

Time Complexity: O(n)

Space Complexity: O(n) due to recursion call stack