Linked List Data Structure

A **linked list** is a fundamental data structure in computer science. It consists of nodes where each node contains data and a reference (link) to the next node in the sequence. This allows for dynamic memory allocation and efficient insertion and deletion operations compared to arrays.

Find Length of a Linked List (Iterative and Recursive)

An iterative approach for finding the length of the linked list:
Follow the given steps to solve the problem:
- Initialize count as 0
- Initialize a node pointer, current = head.
- Do following while current is not NULL
  - current = current -> next
  - Increment count by 1.
- Return count

A recursive approach for finding the length of the linked list:
Follow the given steps to solve the problem:
- If the head is NULL, return 0.
- Otherwise, return 1 + getCount(head->next)

Below is the implementation of the above approach:

Major differences between array and linked-list are listed below:

The major difference between Array and Linked list regards to their structure. Arrays are Index based data structure where each element associated with an index. On the other hand, Linked list relies on references where each node consists of the data and the references to the previous and next element.

There are four key types of linked lists:

- Singly linked lists.
- Doubly linked lists.
- Circular linked lists.
- Circular doubly linked lists.