

LISTS

2024

Aula 02

DEFINITION

01. What are lists?

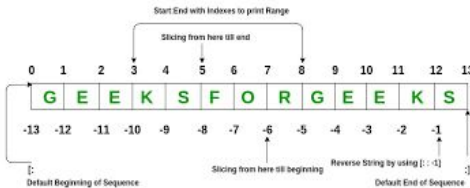
A list is a sequence of several variables, grouped together under a single name.

02. Is it a data structure?

Yes! An array is a linear data structure.

03. How it works?

Arrays use an index system to access elements, where each element has an index number that indicates its position in the array.



COMMON OPERATIONS

- Insertion
- Deletion
- Access
- Slicing
- Traversal
- Searching
- Sorting
- Reversing
- Merging
- Clearing
- Length

ALGORITHM TO LOOP THROUGH A LIST

- Initialize: Start with the list you want to loop through.
- Loop: Use a loop structure to iterate through each element.
- Process: For each element, perform the desired operation (e.g., print, modify, etc.).
- End: Once all elements have been processed, exit the loop.

USE CASE

- Storing Collections of Items: Use lists to hold multiple values, such as a collection of names, numbers, or objects.
- Queue and Stack Implementations: Implement FIFO (first-in-first-out) queues or LIFO (last-in-first-out) stacks using lists.
- Data Manipulation: Easily add, remove, or modify elements, making lists suitable for dynamic data handling.
- Iterating Over Data: Loop through lists to process or analyze each item, often used in algorithms and data processing tasks.

LINKED LIST

2024

Aula 02

DEFINITION

01. What are linked lists?

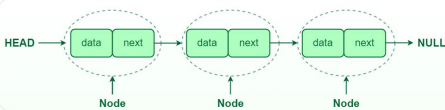
A linked list is a linear data structure like arrays. whereas, linked list do not have elements stored in contiguous locations, as arrays do. In simple words, we can say, linked list is a collection of nodes.

02. What node is consist of?

Each node in the linked list contains the data part and the pointer to the next node in the linked list.

03. Basic Terminologies of Linked List

Head and Tail: The linked list is accessed through the head node, which points to the first node in the list. The last node in the list points to NULL or nullptr, indicating the end of the list. This node is known as the tail node.



TYPES

- Single-linked list
- Double linked list
- Circular linked list

Advantages Of Linked List:

- Dynamic data structure
- No memory wastage
- Implementation
- Insertion and Deletion Operations
- Flexible
- Efficient for large data
- Scalability

COMMON OPERATIONS

- Insertion
- AddFront
- AddBack
- RemoveFront
- RemoveBack
- Traverse
- FrontSize

Time complexity

Operation	Big-O	Note
Access	$O(n)$	
Search	$O(n)$	
Insert	$O(1)$	Assumes you have traversed to the insertion position
Remove	$O(1)$	Assumes you have traversed to the node to be removed