Anas Mustafa Nore

ID:231504891

HASAN KALYONCU ÜNİVERSİTESİ

Table of Contents

DATA STRUCTURE

Definition

A data structure is a systematic way of collecting, organizing, and storing data so that it can be accessed and manipulated efficiently.

Type of data structure

Primitive Data Types

Basic building blocks used to store simple values:

- **int** – Integer numbers
- **char** – Single characters
- **float** – Floating-point numbers (decimal numbers)
- **double** – Double-precision floating-point numbers
- **boolean** – True or False values
- **string** – Sequence of characters

Non-Primitive Data Types

More complex structures built using primitive types. They can be **linear** or **non-linear**.

1. **Linear Data Structures**

Elements are arranged sequentially:

- **Array** – A collection of elements of the same type stored in contiguous memory
- **Linked List** – A sequence of nodes where each node points to the next
- **Stack** – Last-In-First-Out (LIFO) structure
- **Queue** – First-In-First-Out (FIFO) structure

2. **Non-Linear Data Structures**

Elements are connected in a hierarchical or networked way:

- **Tree** – Hierarchical structure with a root node and child nodes
- **Graph** – Collection of nodes (vertices) connected by edges

LINKED LIST

Definition

A linked list is a linear data structure consisting of a sequence of elements called **nodes**, where each node contains data and a reference (or pointer) to the next node in the sequence.

Linked list vs Array

ARRAY

- **FIXED SIZE**
- Difficult to add or remove elements
- Must reserve consecutive memory locations; access to elements is fast and in constant time

LINKED LIST

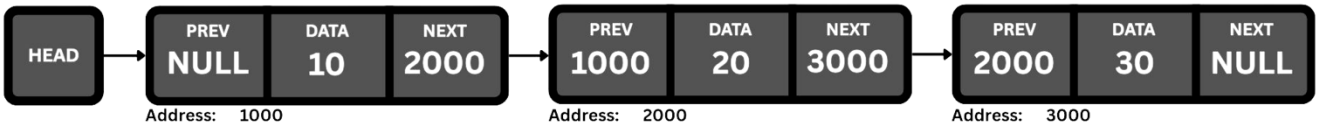
- **FLEXIBLE SIZE**
- Easy to add or remove elements
- Consecutive memory locations are not necessary; access to elements is slower and in non-constant time

Types of Linked List

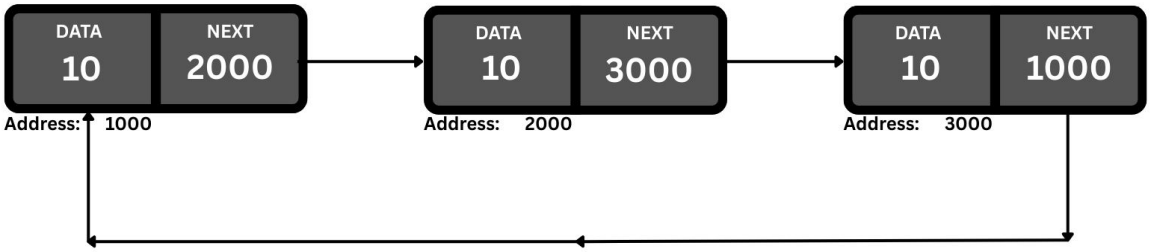
1. **Singly Linked List**:
   - Each node points only to the next node.
   - Can move only forward.
2. **Doubly Linked List**:
   - Each node contains a pointer to the next node and a pointer to the previous node.
   - Can move forward and backward.
3. **Circular Linked List**:
   - The last node points to the first node, forming a circular structure.
   - Can exist in singly or doubly linked forms.

Singly Linked List

Doubly Linked List



Circular Linked List



Basic Operations

1. **Insertion**:
   - At the beginning (Head)
   - At the end (Tail)
   - After a specific node
2. **Deletion**:
   - From the beginning
   - From the end
   - Specific element
3. **Traversal**:
   - Go through all elements to print or search for a value

STACK

Definition

A **Stack** is a data structure that follows the **LIFO (Last In, First Out)** principle.

The last element inserted is the first one to be removed.

Basic Operations

1. **Push**: Insert an element into the stack.
2. **Pop**: Remove the last inserted element.
3. **Peek / Top**: Show the top element without removing it.
4. **isEmpty:** Check if the stack is empty.