**INTRODUCTION**

# What is Data structure?

- A data structure is a way to organize and store values (data) so they can be easily accessed, modified, or removed.

- Data structures can have:

* Relationships between the values (e.g., parent-child in trees).
* Functions or operations associated with them (e.g., inserting or searching).

## Trade-Offs in Data Structures

- No single data structure is universally "best."

- Each has trade-offs between:

* Readability: How easy it is to understand.
* Memory usage: How much space it consumes.
* Speed: How fast it performs operations like searching, inserting, or deleting.

- Choosing the right data structure depends on your specific problem and requirements.

# How computer store data

## Why are Data Structures Important?

- Efficiency: Data structures help the CPU work faster by minimizing the operations needed to access or modify data.

- Scenario-Based: Each data structure is suited for specific use cases, and choosing the right one makes our programs efficient.

## Connection to Hardware

- RAM (Random Access Memory):

* Temporary memory where programs store data while running.
* Data is lost when the computer shuts down.
* Advantage: Fast random access to any memory location.
* Organized in numbered shelves called addresses, with each shelf holding a byte (8 bits).

- Storage:

* Persistent memory (e.g., SSDs, HDDs) used to store data permanently.
* Slower than RAM but retains data after shutdown.

- CPU:

* The "brain" of the computer that processes instructions.
* Relies on RAM for quick data access.
* Uses a cache for even faster access to frequently used data.

A screen shot of a computer

Description automatically generated

Slide 1: CPU & RAM

## Bits and Bytes

- Computers store data in binary (1s and 0s).

- Bit: Smallest data unit (0 or 1).

- Byte: 8 bits.

- Larger systems (e.g., 32-bit or 64-bit) can handle more data, enabling storage of larger numbers or more complex structures.

A white paper with numbers and a red and green square

Description automatically generated

Slide 2: Bít and bytes

## Data Structures in RAM

- Arrangement: Determines how data is stored and accessed in memory.

* Contiguous Memory: Data stored sequentially, enabling fast access (e.g., arrays).
* Non-Contiguous Memory: Data spread across different locations, sometimes slower but offers flexibility (e.g., linked lists).