Day 1 : Data Structures & Algorithms (DSA) with Java – Full Course

Introduction to DSA

@drnimishadavis

Learning Objectives:

By the end of this lesson, you will be able to:

- Understand what DSA is and why it is important.
- Identify different types of data structures.
- Know the difference between data types and data structures.
- Explore real-world applications of data structures.

@drnimishadavis

What is DSA?

DSA stands for Data Structures and Algorithms.
It is one of the most important areas in computer science and programming.

- Data Structures deal with how data is organized, stored, and accessed.
- Algorithms deal with how operations are performed on that data to solve a particular problem efficiently.

In simple terms, data structures help you store data, and algorithms help you process it efficiently.
Together, DSA helps you write optimized programs that use less time and memory.

@drnimishadavis

Example

Suppose you want to store 1000 student names.
You can use:

- An Array if you know the number of names in advance.
- A Linked List if you need to frequently add or remove names.
- A Hash Table if you want to search by student ID quickly.

Choosing the right data structure improves performance drastically.

Why Learn DSA?

1. Efficient Programming: Helps you write faster and more optimized code.
2. Problem Solving: Teaches logical and analytical thinking.
3. Interviews and Exams: Every top tech company (Google, Amazon, Infosys, etc.) asks DSA problems.
4. Foundation of Computer Science: Used in operating systems, databases, AI, and more.

What is a Data Structure?

1. A data structure is a way to organize and store data so that it can be used efficiently.
2. Just like tools in a toolbox are organized for easy access, data structures help programmers organize data for faster operations.

@drnimishadavis

Classification of Data Structures

Data structures are mainly divided into two types:

1. Primitive Data Structures
2. Non-Primitive (or Composite) Data Structures

1. Primitive Data Structures

These are the basic data types that store simple values.
Examples:

- int → Integer numbers (e.g., 10, 25, -7)
- float → Decimal numbers (e.g., 3.14, 9.8)
- char → Characters (e.g., ‘A’, ‘Z’)
- boolean → True or False values

@drnimishadavis

2. Non-Primitive (Composite) Data Structures

These are more complex and can hold multiple values.

They are divided into two categories:

- Linear Data Structures
- Non-Linear Data Structures

@drnimishadavis

Linear Data Structures

In linear structures, elements are arranged in a sequence, one after another.
Examples:

- Array – Stores elements in contiguous memory locations.
- Stack – Follows LIFO (Last In First Out) principle. Example: Undo feature in applications.
- Queue – Follows FIFO (First In First Out) principle. Example: Print queue or ticket counter.
- Linked List – Nodes connected by pointers. Example: Music playlist where songs can be added or removed easily

Non-Linear Data Structures

In non-linear structures, data is not arranged sequentially.
 Elements can be connected in a hierarchical or network form.
Examples:

1. Tree – Hierarchical structure with a root node and child nodes. Example: Folder structure on your computer.
2. Graph – Represents relationships among nodes. Example: Social networks, maps, and recommendation systems.

What is an Algorithm?

An algorithm is a step-by-step method for solving a problem.
Example: To search for an element in a list, you can use:

- Linear Search (check each element one by one)
- Binary Search (check middle element, then divide and search)

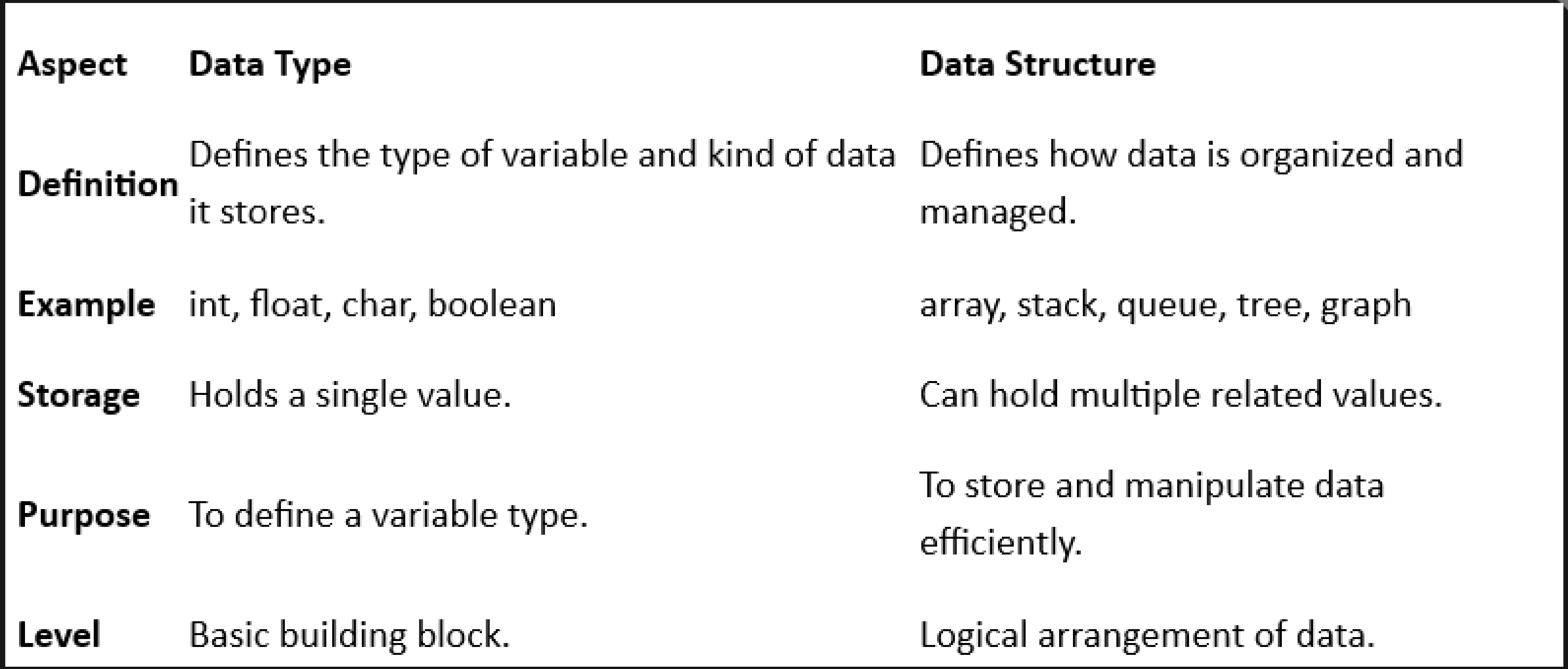
Both work, but Binary Search is much faster for sorted data.
This shows the importance of choosing the right algorithm

Characteristics of a Good Algorithm

1. Input: Should accept zero or more inputs.
2. Output: Should produce at least one output.
3. Definiteness: Each step must be clear and unambiguous.
4. Finiteness: Must terminate after a finite number of steps.
5. Effectiveness: Each step should be simple enough to be executed manually if needed.

@drnimishadavis

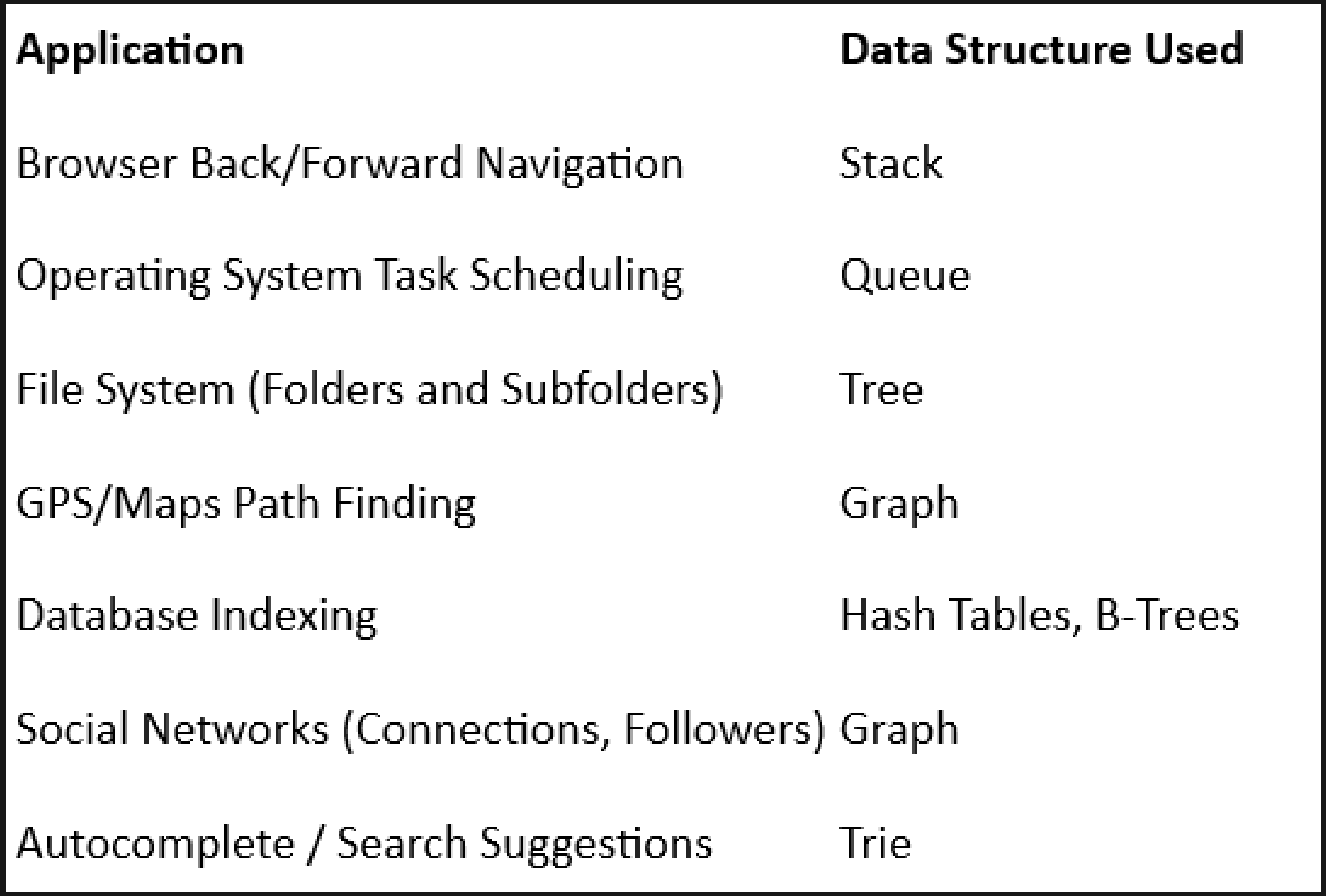
Difference Between Data Type and Data Structure



Aspect	Data Type	Data Structure
Definition	Defines the type of variable and kind of data it stores.	Defines how data is organized and managed.
Example	int, float, char, boolean	array, stack, queue, tree, graph
Storage	Holds a single value.	Can hold multiple related values.
Purpose	To define a variable type.	To store and manipulate data efficiently.
Level	Basic building block.	Logical arrangement of data.

@drnimishadavis

Real-World Applications of Data Structures



Application	Data Structure Used
Browser Back/Forward Navigation	Stack
Operating System Task Scheduling	Queue
File System (Folders and Subfolders)	Tree
GPS/Maps Path Finding	Graph
Database Indexing	Hash Tables, B-Trees
Social Networks (Connections, Followers)	Graph
Autocomplete / Search Suggestions	Trie

@drnimishadavis

Summary

- DSA = Data Structures + Algorithms
- Data structures organize data efficiently.
- Algorithms define the steps to process data.
- Choosing the right data structure leads to better performance.
- DSA knowledge is essential for software development, problem-solving, and placements.

Mini Assignment

- Define DSA in your own words.
- Give two real-life applications of data structures.
- Explain the difference between arrays and linked lists.

@drnimishadavis

Thank You

@drnimishadavis