MIVA OPEN UNIVERSITY

Course:

DATA STRUCTURES

Title:

Week 1 - Summary

School of Computing

COURSE

Data Structures

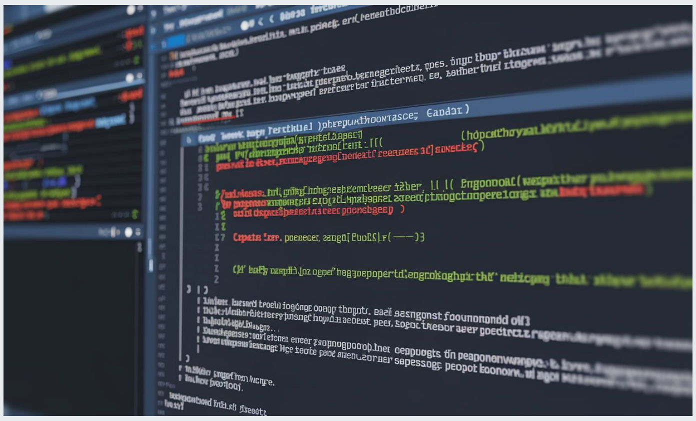
TITLE

Introduction to Data Structures

WEEK 1

Bullet point summary

- Data structures are essential in organising and accessing data efficiently. They help optimise the performance of programs by determining how data is stored in memory.
- Arrays store data in a fixed, ordered way and allow quick access via indices.
- Linked Lists are dynamic and allow for efficient insertion and deletion, but elements are not stored in contiguous memory.
- Stacks follow Last-In-First-Out (LIFO) order and are used in recursion and undo operations.
- Queues operate on a First-In-First-Out (FIFO) basis and are used in task scheduling and managing queues in systems like ticketing or print jobs.
- Trees represent hierarchical data structures, perfect for applications like file systems or organising products in categories.
- Hash Tables map keys to values, providing fast access to specific records, ideal for lookups based on unique identifiers.
- Consideration should be given to performance optimisation, such as retrieval speed, memory management, and ease of data manipulation.



- Understanding the problem at hand is crucial to selecting the appropriate data structure.
- Arrays are used in storing lists like student scores or daily temperatures.
- Linked lists are used in applications like music playlists or undo functionality in word processors.
- Stacks are ssential in managing recursive calls, expression evaluation, and undo operations in software.
- Queues are utilised in managing task scheduling, network traffic, and customer service lines.
- Trees are applied in hierarchical data storage like file systems or organisational charts.
- Graphs are used in modelling networks (e.g., social media, transportation systems) and finding the shortest paths in algorithms.

Key references:

Books Davidson, 2004, Data Structures (Principles and Fundamentals), Dreamtech Press.

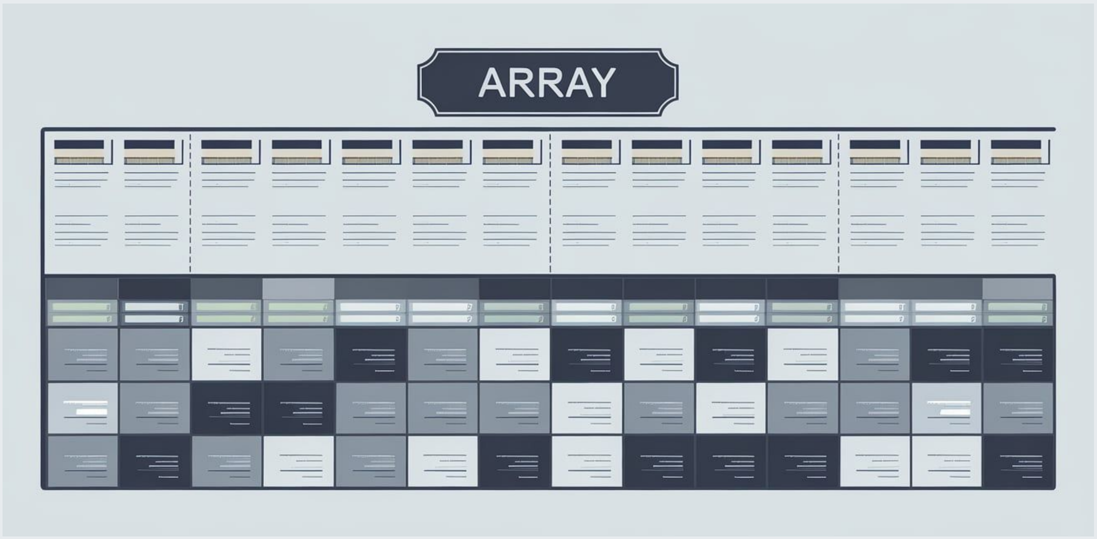
WEEK 1

Personal Reflection

Before proceeding with the next section, think about the following questions to help reflect on your understanding:

- Reflect on how the organisation of data within a program can impact its overall performance and speed.
- Which data structure would you use to manage large sets of user data in a social media app, and why?
- Consider scenarios where linked lists might be more advantageous than arrays, especially when managing dynamic data.

CALL TO ACTION



- Review the various data structures introduced in this lesson and experiment with coding examples in C++.
- Try implementing an array, linked list, stack, or queue in a small project to better understand their functionality.
- Participate in forums or coding platforms to discuss your understanding of data structures with peers and learn from their experiences.

Key references:

Karthikeyan, Fundamentals, Data Structures and Problem Solving, PHI Learning Pvt. Ltd.
Samir Kumar Bandyopadhyay, 2009, Data Structures using C, Pearson Education India.

COURSE

Data Structures

TITLE

Introduction to Data Structures

MIVA OPEN UNIVERSITY

WEEK 1

Questions

1. Why is it important to choose the correct data structure for a given task, and how can it impact the performance of a program?
2. How do dynamic data structures like linked lists compare to static ones like arrays in terms of memory usage and flexibility?
3. What challenges might arise when implementing trees or graphs in real-world applications, and how can they be overcome?
4. In what ways can new technologies and data structures help improve current methods of handling large-scale data in industries like banking or e-commerce?

Skills and Competencies you have acquired after this lesson:

- Problem-solving with data structures
- Efficient data management
- Memory and performance optimisation

Key references:

Pai, G. a. V. (2008). Data Structures and Algorithms.
Sartaj Sahni, 1976, Fundamentals of Data Structures, Computer Science Press.