Chapter 1
Fundamentals of Data Structures:
Stacks, Queues, Linked Lists, and Graphs

D. Varshaa
Coimbatore Institute of Technology, India

A. Keerthana Devi
Coimbatore Institute of Technology, India

M. Sujithra
Coimbatore Institute of Technology, India

ABSTRACT

Computer science encompasses various subfields, with data structures and algorithms being important. This chapter aims to enhance proficiency in constructing and analyzing data structures effectively. Studying data structures equips students with the skills to solve real-world problems using suitable structures. The study of data structures offers diverse programming techniques for complex problems yet introduces challenges due to intricate complexities that reshape program architecture. Covering all aspects in a single semester is impractical, as data structures receive extensive attention in graduate, upper-division, and lower-division programs. This text is an introductory resource for the stack, queue, linked list, graph, trees, searching, and sorting algorithms. It also offers insights into their Python implementation and its complexities, applications, and sample code.

DOI: 10.4018/978-1-6684-7100-5.ch001

INTRODUCTION

A data structure is a meticulously designed framework for organizing, manipulating, retrieving, and storing data. It encompasses a variety of simple and intricate forms, all devised to arrange data in a manner suited to specific use cases. Users can conveniently access and employ the required information using data structures. These structures facilitate data organization in a manner comprehensible to machines and individuals. A data structure can be either developed or selected to facilitate data storage for various operations. In certain instances, the core operations of an algorithm and the design of the data structure are closely intertwined. Each data structure incorporates details about the values and connections of the data, and in rare cases, it may include functions capable of modifying the data.

Conventional elementary data types, such as integers or floating-point numbers, often prove inadequate for effectively expressing the logical intent underlying data processing and application. However, programs involved in receiving, manipulating, and outputting information must be aware of how data should be organized to streamline the processing tasks. Data structures facilitate efficient utilization, persistence, and sharing by logically amalgamating individual data elements. They provide a formal model that delineates the arrangement of data elements. Data structures serve as the foundation for increasingly complex applications. These structures are constructed by aggregating data elements into logical units that represent abstract data types relevant to the algorithm or application at hand, thereby enabling the development of sophisticated systems.

STACK

A stack is a linear data structure that embodies the Last-In-First-Out (LIFO) principle (Carullo, 2020). Under this principle, the item most recently added to the stack is the first to be removed. The stack can be likened to a collection of plates, wherein the plate most recently placed on top is the one to be taken when selecting. Moreover, accessing the plate at the bottom necessitates removing all the plates above it. The stack data structure operates similarly, adhering to this analogy.

Operations On Stack

The following are vital operations associated with a stack data structure:

- **Push:** This operation involves adding an element to the top of the stack.
- **Pop:** This operation entails removing an element from the top of the stack.