

Introduction

1

Outline

- Abstract Data Types
- What is Data Structures and Algorithms all about?

2

Abstract Data Type

3

Primitive Data Types

- A **data type** is characterized by:
 - a domain of *values*
 - a set of *operations* (e.g. *add*, *subtract*, *square root*), which can be applied uniformly to all these values
- Examples: int, float, double, char, string
- Data types depend on the programming language

4

Abstract Data Types

- An **Abstract Data Type (ADT)** is:
 - a set of *values*
 - a set of *operations*, which can be applied uniformly to all these values, but no implementation details.
- ADT defines purely the **behaviour** → what operations can be performed on the set of values, rather than how it is implemented.
- Examples:
 - List
 - Stack
 - Queue
 - Tree

5

List ADT - Example

- List of values
- Operations:
 - Insert
 - Delete
 - Order

6

Queue ADT - Example

- Operates in a FIFO (First In First Out) manner
- Queue operations
 - create
 - destroy
 - enqueue
 - dequeue
 - is_empty



7

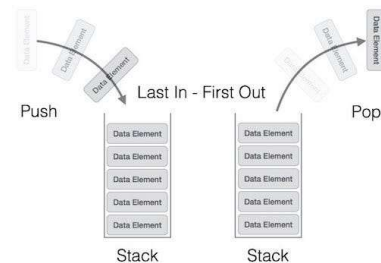
Applications of the Queue

- Hold jobs for a printer
- Store packets on network routers
- Make waitlists fair
- Breadth first search

8

Stack ADT

- A last-in, first-out (LIFO) structure
- Operations happen only on one end (the top)
- Operations: Push, pop, peek
- Push → Insert data element to the top
- Pop → Remove data element from the top
- Peek → See what is on top



9

Abstract Data Types

- An ADT couples its data (the set of values) and operations (methods/ functions)
- From Latin abstract means to 'pull out' the essentials → to defer or hide the details
- Abstraction emphasizes essentials and defers/hides the details
- ADT only mentions what operations are to be performed but not how these operations will be implemented and is independent of any programming language → Think of it as from a user point of view and not a programmer.

10

What is Data Structures and Algorithms all about?

11

Data Structures and Algorithms

- Data Structure:
 - It is a way of organizing and storing data so that it can be accessed and updated efficiently.
 - They contain operations (implemented) to manipulate data elements.
 - ADT is in the logical level while a data structure is in the implementation level.
 - Data structures implement ADTs.

12

Data Structures and Algorithms

- Data Structure:
 - A list can be described in terms of an abstract data type (we can insert into it, get the nth element, delete an element, etc.),
 - A linked list which is a data structure, for example, is an implementation of a list abstract data type → it implements the specified behavior by, for example, by providing functions for inserting an element, deleting an element, getting the nth element.
 - We could implement the same list abstract data type in many other ways, for example with an array, or binary tree.
 - Queue:
 - FIFO ADT
 - Can be implemented using array or linked list data structures

13

Data Structures and Algorithms

- Algorithm:
 - A high level, language independent description of a step-by-step process for solving a problem

14

Example Data Structures

- Queue
- Stack
- Linked list
- Graphs
- Binary trees

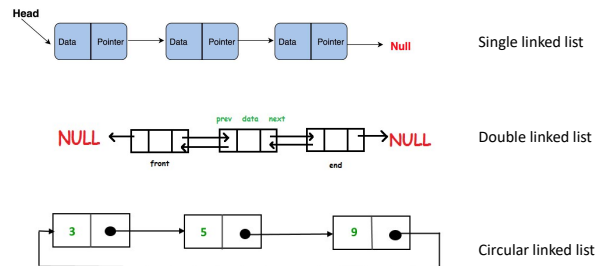
15

Example Algorithms

- Search Algorithms
- Sort Algorithms

16

Example Applications – Linked List



17

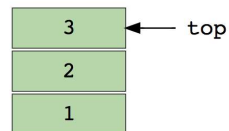
Example Applications – Linked List

- Images are linked with each other. So, an image viewer software uses a linked list to view the previous and the next images using the previous and next buttons.
- Web pages can be accessed using the previous and the next URL links which are linked using a linked list.
- The music players also use the same technique to switch between music.
- It can be used to implement Stacks, Queues, Graphs, and Trees.
- UNDO, REDO or DELETE operations in a notepad (stack)
- Used in switching between applications and programs (Alt + Tab) in the Operating system → circular linked list
- Escalators — double linked List.

18

Stack

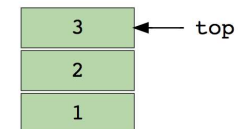
- Stack data structure is one in which you insert and remove from only one end
- Insert → push
- Remove → pop



19

Example Applications - Stack

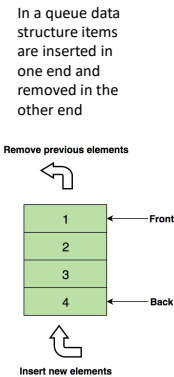
- Undo/Redo button/operation in word processors.
- Forward-backward surfing in the browser.
- Message logs and all messages you get are arranged in a stack.
- Call logs, E-mails, Google photos' gallery, YouTube downloads, Notifications (latest appears first).
- Java Virtual Machine.
- Recursion.



20

Example Applications - Queue

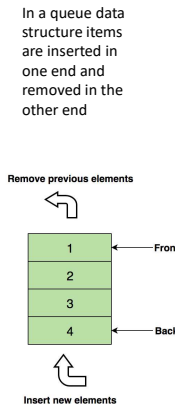
- Operating System uses queues for job scheduling.
- Data packets to a router are arranged in queue format.
- Server while responding to request e.g when sending an e-mail to an email server, it will be queued.
- Uploading and downloading photos, first kept for uploading/downloading will be completed first
- Queueing systems in banks or hospitals
- In escalators and printers



21

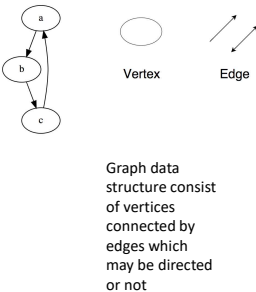
Example Applications - Queue

- Uploading and downloading photos, first kept for uploading/downloading will be completed first
- A queue can be implemented in - Linked List-based Queue, Array-based Queue, Stack-based Queue.



Example Applications - Graph

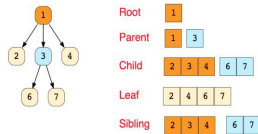
- Dijkstra algorithm or the shortest path first algorithm also uses graph structure to find the smallest path between the nodes of the graph.
- Transportation networks like the one used by Google Map.
- Airline network.
- The GPS navigation system also uses shortest path APIs.
- BFS (Breadth First Search) and DFS (Depth First Search) algorithms



23

Example Application - Tree

- In computer systems, directory and file systems
- Implementation of navigation structure of a website
- To search an element in a set quickly, Binary Search Trees(BSTs) are used.
- Shortest path trees and spanning trees are used in bridges and routers.
- Databases also use tree data structures for indexing.
- Domain Name Server(DNS) also uses tree structures.
- File explorer/my computer of mobile/any computer



24