D. S. with C

Introduction to Data Structure

Course on C-Programming & Data Structures: GATE - 2024 & 2025

Vishvadeep Gothi • Lesson 18 • Dec 16, 2022

Data Structure:
Introduction

By: Vishvadeep Gothi

3rd year ⇒ 682

4th year ⇒ 19 → IISc Bangalore

process ⇒ better rank
of Preparation

119, 440

E_Learning
DS
D.L.
D.M.
math
optim

2018 - 2020 → MTech in D.S. & ML
(BITS Pilani)

Data Structure

Chapter Number	Chapter Name
1	Data Structure Basics
2	Arrays
3	Linked List
4	Queue & Stack
5	Tree
6	Graph
7	Hashing

Expression
Recursion

Heaps

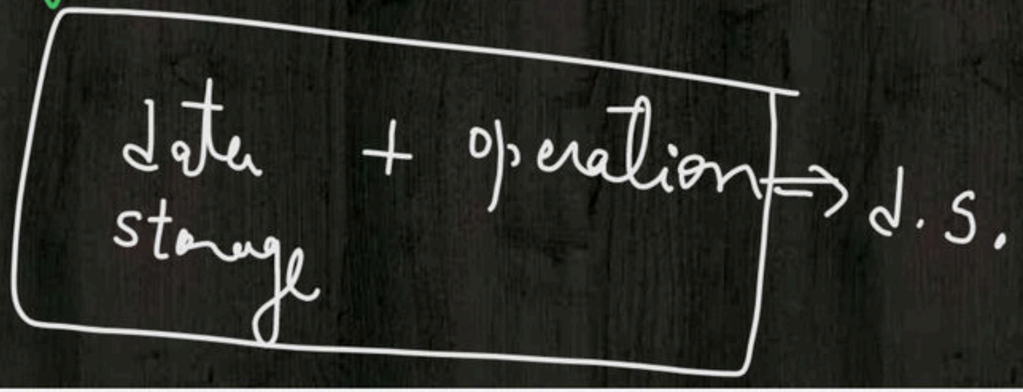
Data Structure

- Mathematical and logical model of organizing the interrelated data.

predefined way arrange or organize data

operations to be performed on data

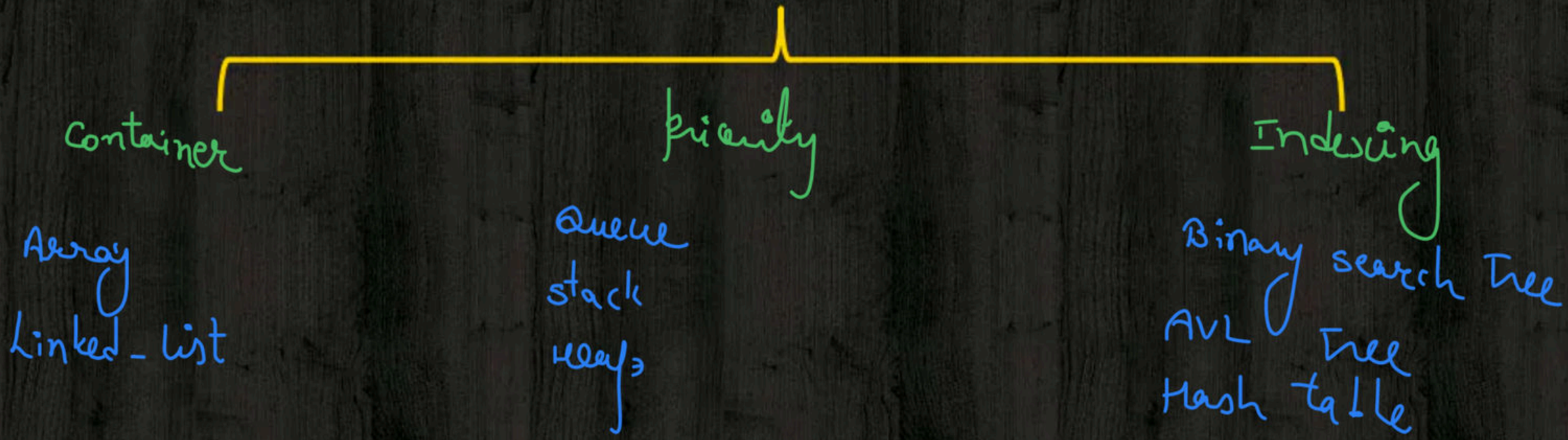
ADT :- Abstract Data Type



Types of Data Structure

- **Types:**
  1. **Linear:** Elements are arranged in linear (sequential) fashion.
     ↳ Array, Linked-list, Queue, stack
  2. **Non-Linear:** Elements are arranged in non-linear fashion.
     ↳ Tree, Heaps, Graph, hash-table

Classification of Data Structure



Why data structures?

To store data in RAM (main memory) in organized manner so that processing can be done in efficient manner by program or algorithm.

Operations on Data Structures

1. **Traversing :-** Visiting each element of data structure exactly once.

2. **Insertion :-** Adding a new element in an existing data structure.

Posibility of an error ⇒ overflow

↳ when trying for insertion is on already full data structure.

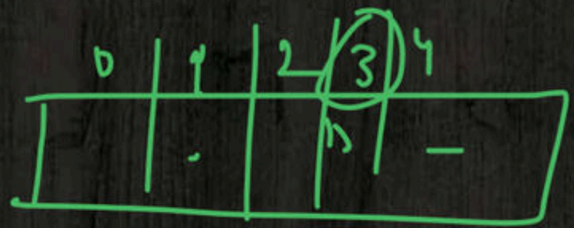
3:- Deletion :- Removing an element from a data structure

Possibility of error => Underflow

↓

when trying for deletion is already empty data structure

Operations on Data Structures



1:- Searching :-

Finding an element is data structure

Linked list or Tree

array

Result → Search successful ⇒ Element present in DS ⇒ Return location of element

index of element

Search unsuccessful ⇒ Element is not present in DS ⇒ return default value for unsuccessful search

-1

	Searching in linked list or Tree
Successful search	Return add. of node
Unsuccessful search	NULL pointer

Operations on Data Structures

5:- Sorting:- Arranging elements in ascending or descending order

6:- Merging:- Combining 2 data structures of similar type, into one.

Algorithm

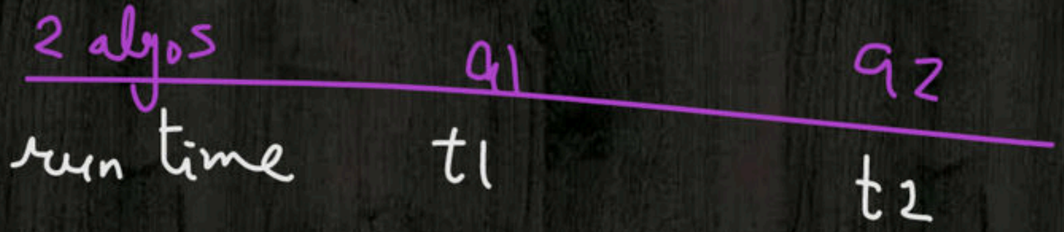
→ step by step solution approach for an algorithmic problem.

Analysis of Algorithm

→ To compare multiple soln^{ns} for the same problem.

Analysis of Algorithm

- **Space Complexity** → Space needed by algorithm to run (apart from input & output)
  ↳ (memory)
- **Run-Time Complexity**
  ↳ time needed by algorithm to run



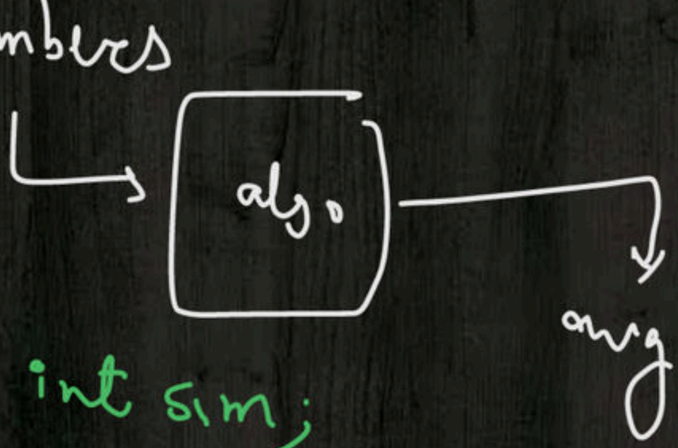
if $t1 > t2$

t2 is efficient.

Analysis of Algorithm

esr:
algo

takes n numbers as input
and returns avg. of n numbers



int sum;

sum = num1 + num2 + ... + numn

avg = sum/n;

⇓

space needed
by algo => 1

Ex 2:-

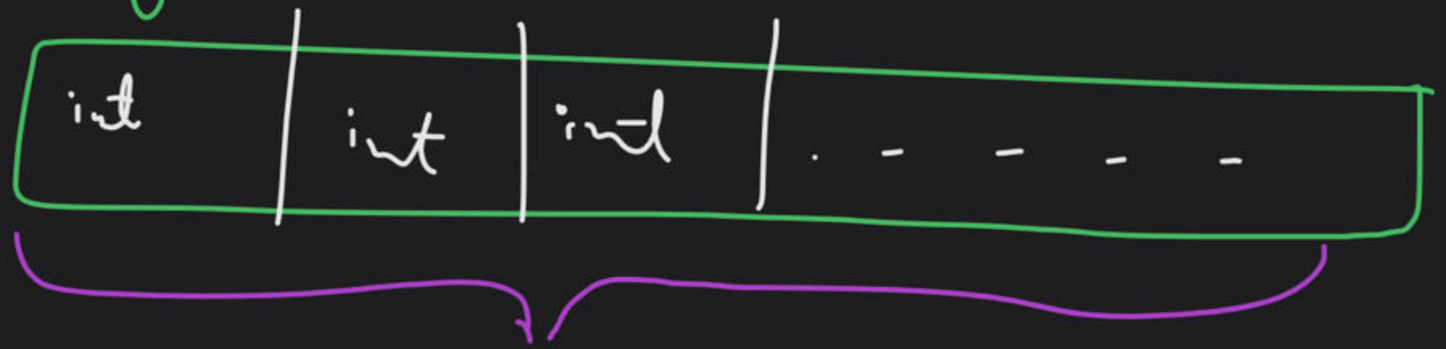
input => n number of float elements

output => sum of only int part of each element.

---

approach:-

- array taken to store only int part



no. of space taken by algo. => n

Example: 1

Example: 2

Example: 3

Asymptotic Notation

Asymptotic Notation

Question 1

Consider an algorithm which takes n number of inputs and performs an operation on it, which requires n-1 operations. The best possible run time complexity for the algorithm can be represented as:

(A) $O(n)$

(B) $\Theta(n)$

(C) $O(n \log_2 n)$

(D) A & B both

Question 2

Consider an algorithm which takes n number of inputs and performs an operation on it. The operation is performed by algorithm in such a way that it is not dependent on number of inputs. Which of the following can be the run time complexity for the algorithm?

(A) O(1)

(B) Θ(1)

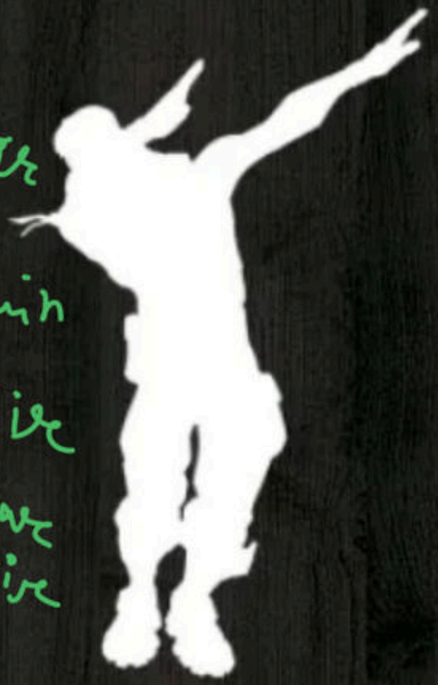
(C) O(n)

(D) All

Happy Learning

C - DS
COA
OS
DBMS

EM :) gurupal sir
shrenik jain
BV Reddy sir
Uma maheshwar sir



[Sanchit sir
Subba rao]

ROR sir
Gunturu sir → Recorded