

# Five Day Faculty Orientation Workshop on Data Structures 2019 Course

**Prof. Digambar Padulkar** Assistant Professor, VPKBIET, Baramati



Organized By

Hope Foundation's

**International Institute of Information Technology, Pune**

**Under the Aegis of BoS (E&TC), SPPU, Pune SE E&TC/ Electronics) 2019 Course**

22<sup>nd</sup> to 26<sup>th</sup> June 2020



**Course Code: 204184**

**Course: Data Structures**

Program	Class	Academic Year
UG Program in EnTC/Electronics	SE (EnTC/Electronics)	2020-21 Sem-I

**Corresponding Laboratory: Data Structures Laboratory(204188)**

# Data Structures-Teaching & Examination Scheme

## Teaching Scheme

Theory hrs/week	Practical hrs/week	Tutorial hrs/week
3hrs	2hrs	-

## Examination Scheme

Theory			Lab		
InSem	EndSem	Sessional	TW	PR	OR
30	70	-	-	-	25

# Delivery Methods

Chalk & Talk	ICT Tools	Group Discussion	Industrial Visit	Expert Talk	Animation & Role Play	Mini Project	Lab
✓	✓			✓	✓	✓	✓

- 1 To learn basic concepts of C Programming language.
- 2 To learn different sorting and searching algorithms and their analysis.
- 3 To learn linear data structures: Stack, Queue, Linked List and their applications.
- 4 To learn nonlinear data structures: Tree, Graph and their applications.
- 5 To study the systematic ways of solving problem, various methods of organizing large amount of data.
- 6 To solve problems using data structures such as binary tree, binary search tree & graph and writing programs.

# CO-PO Mapping

Course Outcome	RBT Level	Student will be able to	Mapping with Unit	PO Mapping
CO204184.1	2	Develop programs using C programming language	1	
CO204184.2	4	Implement sorting and searching algorithms and calculates its complexity	2	1,2,4,12
CO204184.3	3	Develop applications of stacks and queues using array	3	
CO204184.4	3	Demonstrate applicability of linked list	4	
CO204184.5	3	Demonstrate applicability of nonlinear data structure – Binary Tree with respect to its time complexity	5	
CO204184.6	4	Apply the knowledge of graph for solving the problems of spanning tree and shortest path algorithm	6	

# CO-PO Mapping and Justification

CO2-PO1	2	Writing program itself is a practical representation of theoretical mathematics
CO2-PO2	3	Applying knowledge of mathematics, write a programs and analyse for timing requirement
CO2-PO4	3	Applying well know searching and sorting methods on set of data to understand, if whole data is unsorted, partially sorted and sorted
CO-PO12	2,3	Programming is a life long learning scheme

# Searching and Sorting

## Searching and Sorting

**Algorithms:** Analysis of Iterative and Recursive algorithms, Space & Time complexity, Asymptotic notation- Big-O, Theta and Omega notations.

**Searching methods:** Linear, Binary and Fibonacci Search

**Sorting methods:** Bubble, Insertion, Selection, Merge, and Quick Sort.

# Algorithm-Is it a Program?

## Algorithm

It's collection of finite number of the instructions organized in specific order to get desired output

## It plays a role

Yes, Algorithm plays important role, In understanding the timing analysis-It's completely based upon the what **Data organizing** and **Data Processing** Data Structures we are choosing



# Example Problem

You are given an array with  $n$  integers, task is to identify the number repeats  $\frac{n}{2}$  times

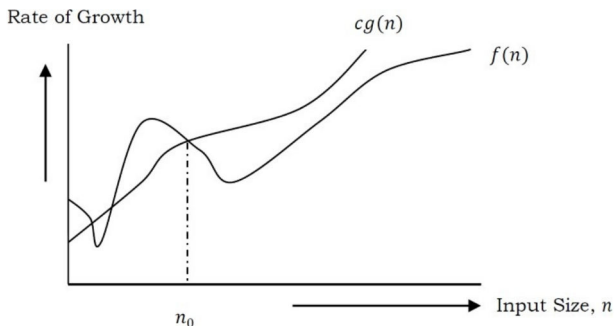
## Identifying need-Decomposing Problem

- Array of integers-Data organizing data structure
- Logic
- Algorithm

# Asymptotic Notations

## Big O

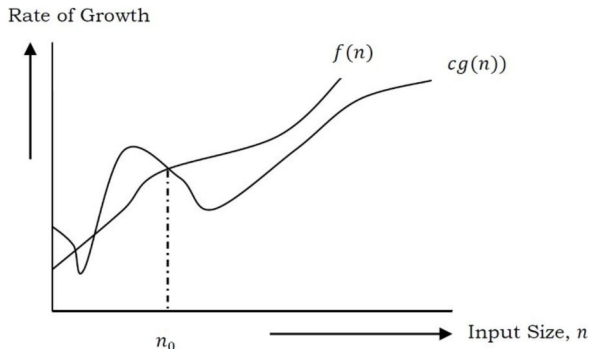
**O (Big Oh):-** The function  $f(n) = O(g(n))$  (read as f of n is big oh of g of n) if and only if there exists positive constants  $c$  and  $n_0$  such that  $f(n) \leq c * g(n)$  for all  $n, n \geq n_0$ .



# Asymptotic Notations

## Omega

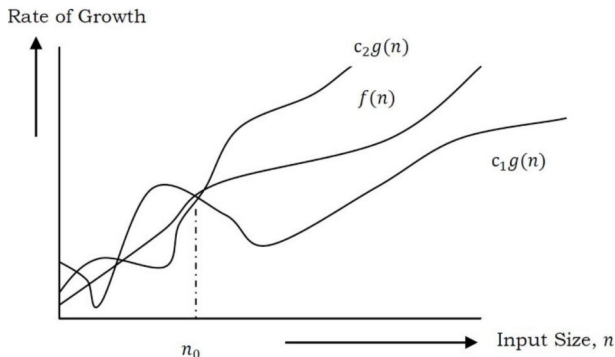
$\Omega$  (**Omega**):- The function  $f(n) = \Omega(g(n))$  (read as f of n is omega of g of n) if and only if there exists positive constants  $c$  and  $n_0$  such that  $f(n) \geq c * g(n)$  for all  $n, n \geq n_0$ .



# Asymptotic Notations

## Theta $\theta$

$\theta$  (**Theta**):- The function  $f(n) = \theta(g(n))$  (read as f of n is theta of g of n) if and only if there exists positive constants  $c_1$ ,  $c_2$  and  $n_0$  such that  $c_1g(n) \leq f(n) \leq c_2g(n)$  for all  $n, n \geq n_0$ .



# Time Complexity

Statement	s/e	Frequency	Total steps
int sequentialSearch(...)	0	0	0
{	0	0	0
int i;	1	1	1
for (i = 0; i < n && x != a[i]; i++);	1	$n + 1$	$n + 1$
if (i == n) return -1;	1	1	1
else return i;	1	0	0
}	0	0	0
Total			$n + 3$

Source:- Data Structures and Algorithms, Sartaj Sahni and Dinesh Mehata, Champman CRC Press

# Finding $cg(n)$

n	f(n)	cg(n)
0	3	0
1	4	2
2	5	4
3	6	6
4	7	8
5	8	10

# Recursive Algorithm-Merge Sort

$$T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + n \quad (1)$$

$$T(n) = 2T\left(\frac{n}{2}\right) + n \quad (2)$$

$$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{4}\right) + \frac{n}{2} \quad (3)$$

Put equation 3 in 2 we have

$$T\left(\frac{n}{2}\right) = 2\left[2T\left(\frac{n}{4}\right) + \frac{n}{2}\right] + n \quad (4)$$

Simplifying earlier equation we get

$$T(n) = 2^2 T\left(\frac{n}{2^2}\right) + 2n \quad (5)$$

Generalizing the equation we have

$$T(n) = 2^k T\left(\frac{n}{2^k}\right) + kn \quad (6)$$

$2^k$  reaching to the maximum  $n$  then

$$T(n) = nT\left(\frac{n}{n}\right) + n \log n \quad (7)$$

Ultimately this  $g(n)$  becomes  $n \log n$



# Iterative and Recursive Algorithms

Iteration and recursion are two fundamental concepts without which it would be impossible to do much, if anything, useful in computing.

## Iterative and Recursive forms of Quick Sort

Quick sort algorithm can be written in recursive or iterative form

## Trees

Trees can be written in iterative and recursive forms

# Searching Algorithms

## Linear Search

Searching an element of our interest into the one dimensional entity. Best example is array

It takes linear growth of algorithm over the input size  $n$

So worst case time becomes  $O(n)$

# Whats our aim

## AIM

We are in search of data structure, which will organize and manage the data in such a way that searching time will be reduced

# Binary Tree

## Binary Tree Organization

Data is organized in tree structure without any constraints, possibility of getting the thing to be searched is equally probable in left and right subtree

We will be getting that element in either left or right part of the tree but we don't have mechanism to choose the right or left

Other variants of binary tree such as binary search have a decision mechanism to go to left or right, so time requirement is going down to  $\log n$  directly

# Sorting Problems

Bubble, Insertion, Selection, Merge, and Quick Sort

# University Question Paper Questions

1

You are given an array of **n** integers 1 to n. Each integer appears exactly once, If **a** and **b** are the any two integers in the given range. Write a algorithm to find out how many number of elements are there in between **a** and **b**

2

You are given an array with **n** numbers, write an algorithm to find out a number which repeats  $\frac{n}{2}$  times

3

Ram and Yusuf wants take a roadtrip from Guwahati to Pune. It's long journey of 62 hours. Given a road map as weighted graph  $G = (V, E, w)$  with only positive weights, here  $V$  represents Halting vertices,  $E$  represents Roads and  $w$  represents positive distance between the cities. Devise an efficient algorithm to find out route with minimum distance from Guwahati to Pune

# TiPs for Question Paper Setting

1

If want to ask question based upon Explain..... with example

Dont ask this kind of the questions, still if you want to ask then give your own example

Try to

Make question paper interesting and free from W's

2

Try to make the questions free from questions



Refer Foreign Universities like Michigan Tech, University of Michigan, North Carolina State University, Iowa State University, Rutgers University, Yale University etc. for how they frame the Assignments, Questions and Tutorials

# Thank You!