



DATA STRUCTURES AND ITS APPLICATIONS

Vandana M L

Department of Computer Science and Engineering

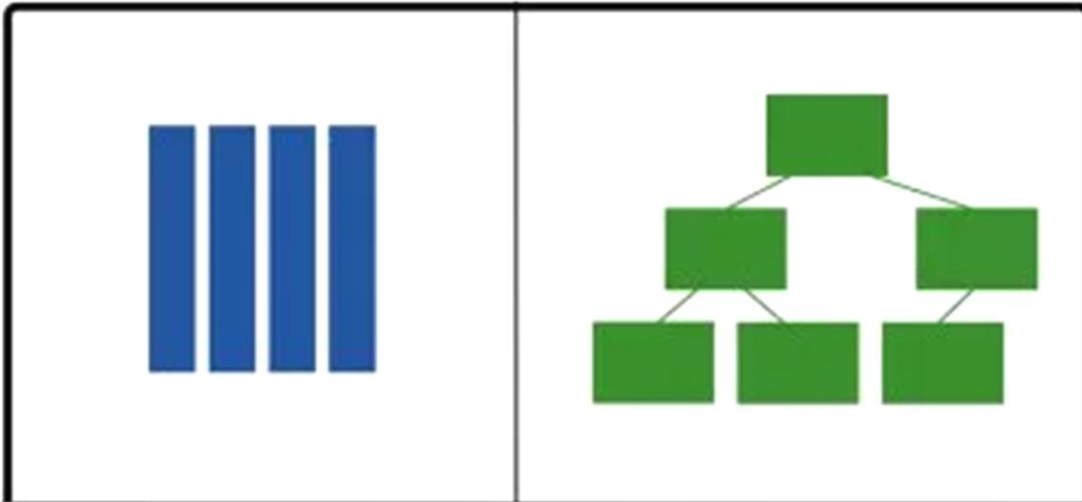
DATA STRUCTURES AND ITS APPLICATIONS

Introduction to Data Structures

Vandana M L

Department of Computer Science and Engineering

Data Structure is a scheme of organizing data in the memory of the computer in such a way that various operations can be performed efficiently on this data



DATA STRUCTURES AND ITS APPLICATIONS

Introduction to Data Structures

Why Data Structure?

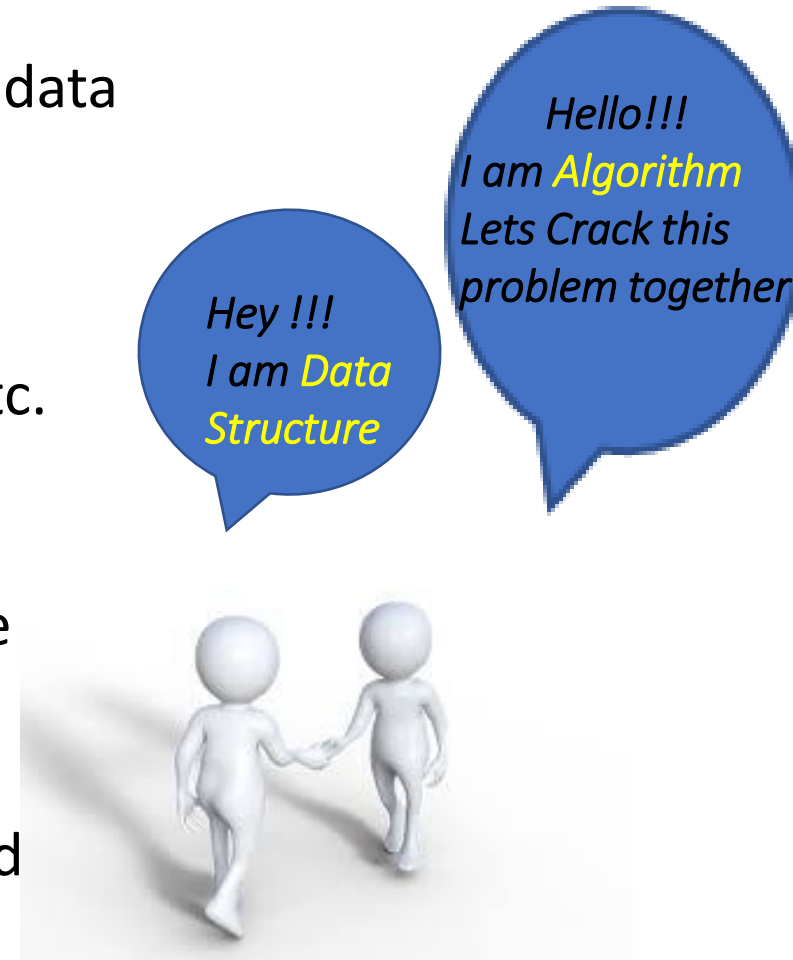


DATA STRUCTURES AND ITS APPLICATIONS

Introduction to Data Structures

Why Data Structures?

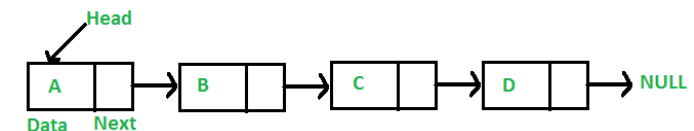
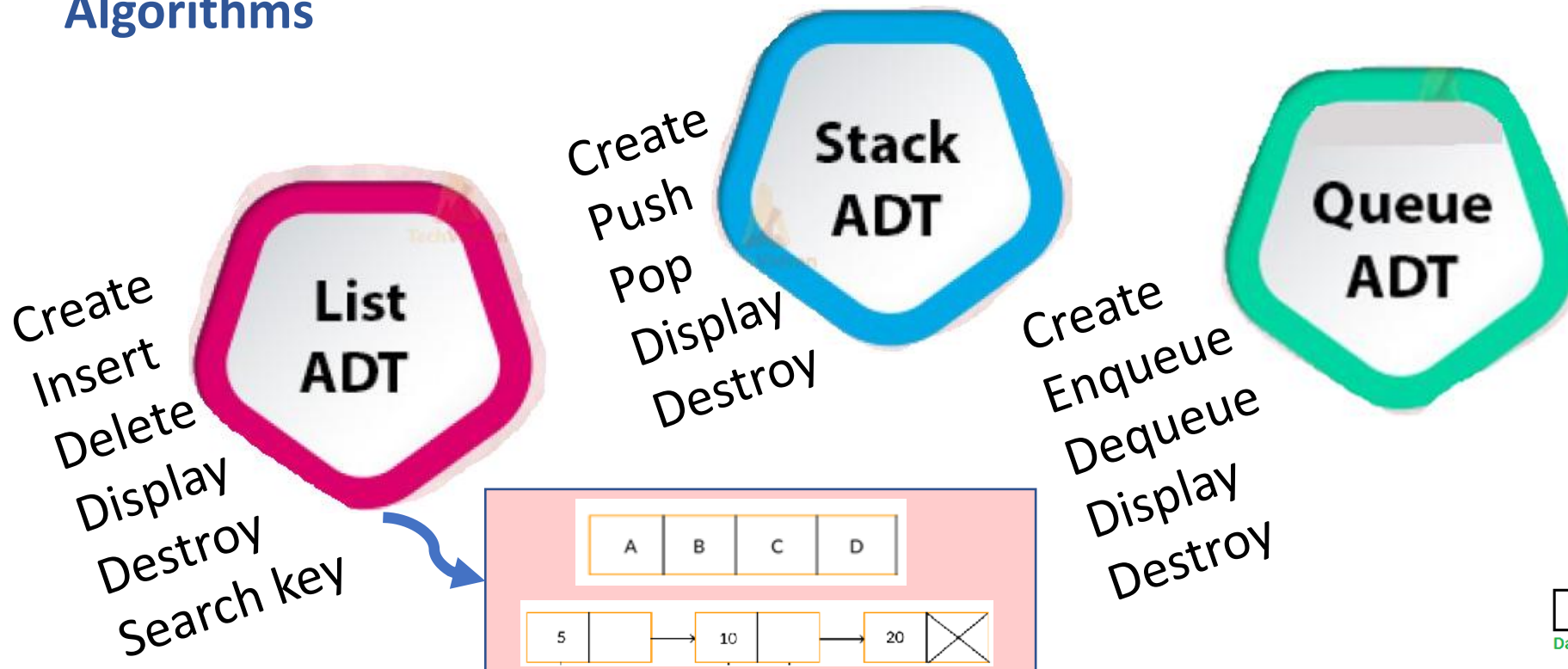
- Computer systems deal with large amount of data (text ,image, relational data etc.)
- Data is just the raw material for information, analytics, business intelligence, advertising, etc.
- The way data is organized in memory plays a key role in deciding the time complexity of the algorithms designed for solving the problems
- Data Structures and algorithm go hand in hand



Importance of Data Structures

- Data Structures is most fundamental and building block concept in computer science
- Good knowledge of Data Structures is required to build efficient software systems

- Abstract Data Type is used to represent data and operations associated with an entity from the point of view of user **irrespective of implementation**
- ADT can be implemented using one or more **Data Structures** and **Algorithms**



➤ Linear Data Structures

Stack, Queue, Linked List

➤ Non Linear Data Structures

Tree , Graph



Linear Organisation

Stack

Queue

Linked List

Linear List using Array



Non Linear Organisation

*Tree
Graph*

DATA STRUCTURES AND ITS APPLICATIONS

Few Applications of Linear Data Structures



➤ Array

- To implement other data structures
- To store files in memory

➤ Linked Lists

- To implement other data structures
- To manipulate large numbers

➤ Stacks

- Recursion
- Infix to postfix conversion

➤ Queues

- Process Scheduling
- Event handling

➤ Tree

- Auto complete features (Trie)
- Used by operating systems to maintain the structure of a file system

➤ Heaps

- Priority Queue implementation
- Heap Sort

➤ Graphs

- Computer Networks
- Shortest Path Problems

Unit -1 : Linked List and Stacks

Review of C , Static and Dynamic Memory Allocation. Linked List: Doubly Linked List, Circular Linked List – Single and Double, Multilist: Introduction to sparse matrix (structure). Skip list Case study: Dictionary implementation using skip list Stacks: Basic structure of a Stack, Implementation of a Stack using Arrays & Linked list. Applications of Stack: Function execution, Nested functions, Recursion: Tower of Hanoi. Conversion & Evaluation of an expression: Infix to postfix, Infix to prefix, Evaluation of an Expression, Matching of Parenthesis.

Unit -2 : Queues and Trees

Queues & Dequeue: Basic Structure of a Simple Queue, Circular Queue, Priority Queue, Dequeue and its implementation using Arrays and Linked List. Applications of Queue: Case Study – Josephus problem, CPU scheduling- Implementation using queue (simple /circular). General: N-ary trees, Binary Trees, Binary Search Trees (BST) and Forest: definition, properties, conversion of an N-ary tree and a Forest to a binary tree. Traversal of trees: Preorder, Inorder and Postorder.

Unit -3 : Application of Trees and Introduction to Graphs

Implementation of BST using arrays and dynamic allocation: Insertion and deletion operations, Implementation of binary expression tree., Threaded binary search tree and its implementation. Heap: Implementation using arrays. Implementation of Priority Queue using heap - min and max heap. Applications of Trees and Heaps: Implementation of a dictionary / decision tree (Words with their meanings). Balanced Trees: definition, AVL Trees, Rotation, Splay Tree, Graphs: Introduction, Properties, Representation of graphs: Adjacency matrix, Adjacency list. Implementation of graphs using adjacency matrix and lists. Graph traversal methods: Depth first search, Breadth first search techniques. Application: Graph representation: Representation of computer network topology.

Unit -4 : Applications of Graphs , B-Trees, Suffix Tree and Hashing

Application of BFS and DFS: Connectivity of graph, finding path in a network. Suffix Trees: Definition, Introduction of Trie Trees, Suffix trees. Implementations of TRIE trees, insert, delete and search operations. Hashing: Simple mapping / Hashing: hash function, hash table, Collision Handling: Separate Chaining & Open Addressing, Double Hashing, and Rehashing. Applications: URLs decoding, Word prediction using TRIE trees / Suffix Trees.

Text Book :

"Data Structures using C / C++", Langsum Yedidiah, Moshe J Augenstein, Aaron M Tenenbaum Pearson Education Inc, 2nd edition, 2015.

Reference Book:

"Data Structures and Program Design in C", Robert Kruse, Bruce Leung, C.L Tondo, Shashi Mogalla, Pearson, 2nd Edition, 2019

Data Structures and its Applications

Evaluation Policy



Pattern:	ISA: 50 marks ESA: 50 marks
ISA 1 / ISA 2:	Hybrid / CBT, conducted for 40 marks, scaled down to 20 marks (each) (16 * 1) mark MCQs + (4 * 2) marks MCQs + (4 * 4) marks descriptive questions. Total: 20 + 20 marks.
Lab Component/FSA:	10 Labs (Implementation based): 10 marks Jackfruit problem (Mini): 10 marks Total : 20 Marks
Assignment / Experiential Learning	Banana problem: 5 marks (pen and paper) (classnotes , HW) Orange problem: 5 marks (Implementation based in assessment center) Total: 10 marks
ESA:	25 marks from each unit, conducted for 100 marks, reduced to 50 marks
ISA + ESA + Lab + Experiential Learning	120 reduced to 100 marks



THANK YOU

Vandana M L

Department of Computer Science & Engineering

vandanamd@pes.edu