Computer Fundamentals

Priti Srinivas Sajja
Professor
Department of Computer Science
Sardar Patel University

Visit pritisajja.info for details

Unit 4: Data Structure

- Introduction
- Types of DT
- Array
- Stack
- Queue
- Linked Lists
- Tree and Graph
- Hashing
- Search & Sort

- Name: Dr. Priti Srinivas Sajja
- Communication:
  - Email : priti@pritisajja.info
  - Mobile : +91 9824926020
  - URL :http://pritisajja.info
- *Academic qualifications* : Ph. D in Computer Science
- *Thesis title*: Knowledge-Based Systems for Socio-
- Economic Rural Development (2000)
- *Subject area of specialization* : Artificial Intelligence
- *Publications* : 216 in Books, Book Chapters, Journals and in Proceedings of International and National Conferences

Unit 4: Data Structure

- **Introduction**
- Types of DT
- Array
- Stack
- Queue
- Linked Lists
- Tree and Graph
- Hashing
- Search & Sort

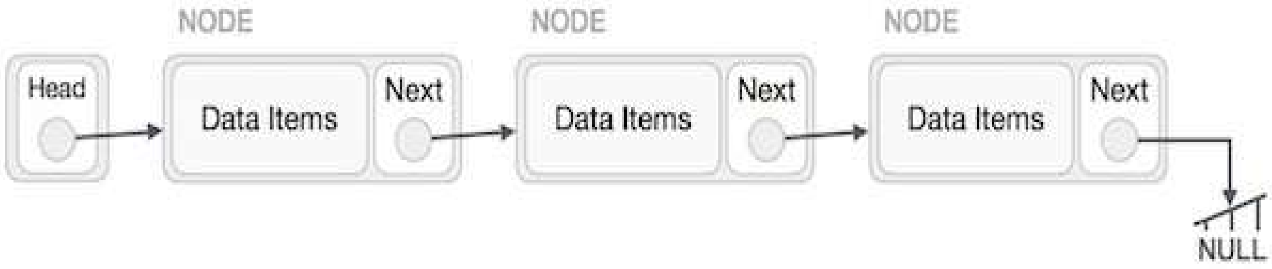
Unit 4: Data Structures

- Primitive and composite data types
- Arrays, stacks, queues, linked lists
- Binary trees, B-trees
- Hashing techniques
- Linear Search, Binary Search
- Bubble Sort

Unit 4: Data Structure

Linked List

- **A linked list is a sequence of data structures, which are connected together via links.**



Basic Operations

- **Insertion** – Adds an element at the beginning of the list.
- **Deletion** – Deletes an element at the beginning of the list.
- **Display** – Displays the complete list.
- **Search** – Searches an element using the given key.
- **Delete** – Deletes an element using the given key.

Unit 4: Data Structure

Introduction

Types of DT

Array

Stack

Queue

Linked List

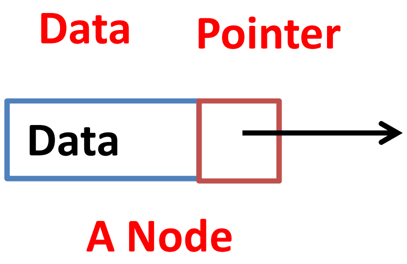
Tree and Graph

Hashing

Search & Sort

Linked List

- Linked list consists of linked nodes.
- Each node is having a data and a pointer part.
- The data part stores data in it.
- The pointer is an address pointing towards the next element of the list.



Data Pointer

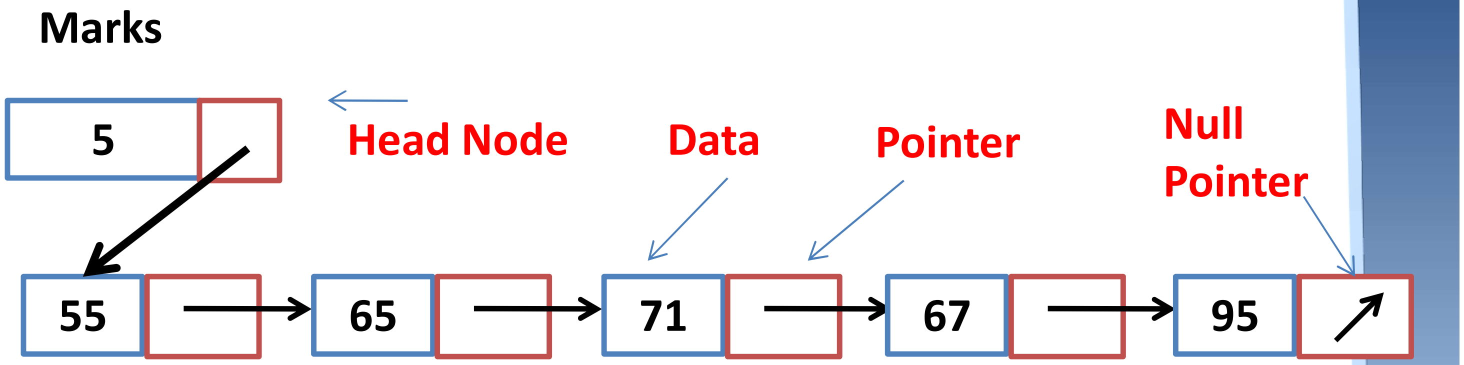
Data

A Node

- Each list is having a head node.
- In a head node data part contains name of list and pointer contains address of the first (next) node of the list.

Linked List Example

- Create a link list of 5 subjects marks of a student called Marks.



Introduction

Types of DT

Array

Stack

Queue

Linked List

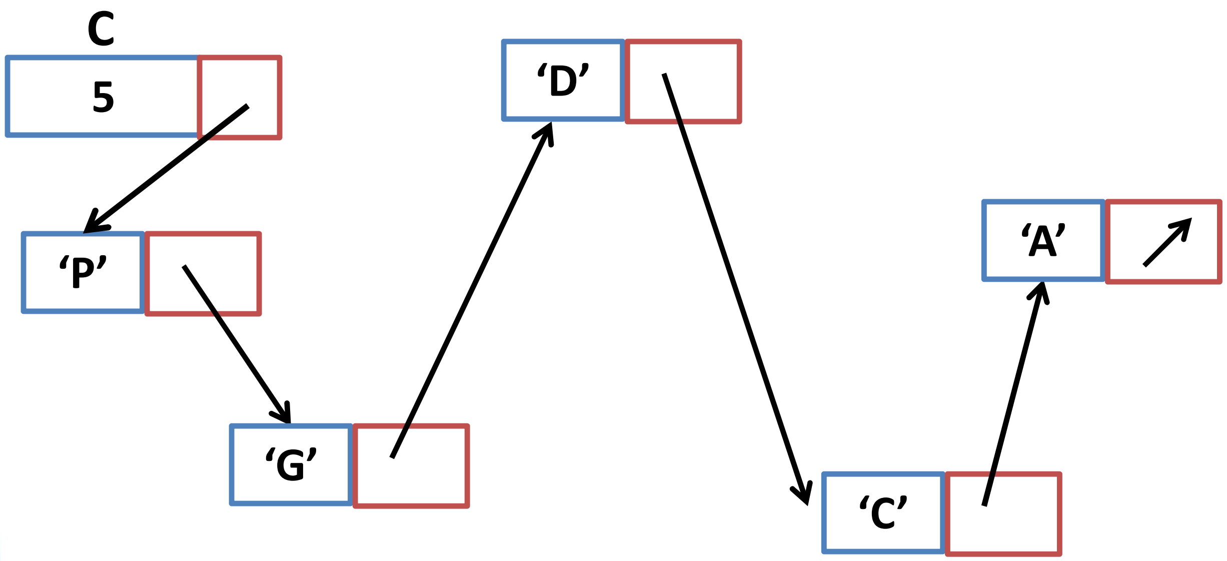
Tree and Graph

Hashing

Search & Sort

Linked List Example

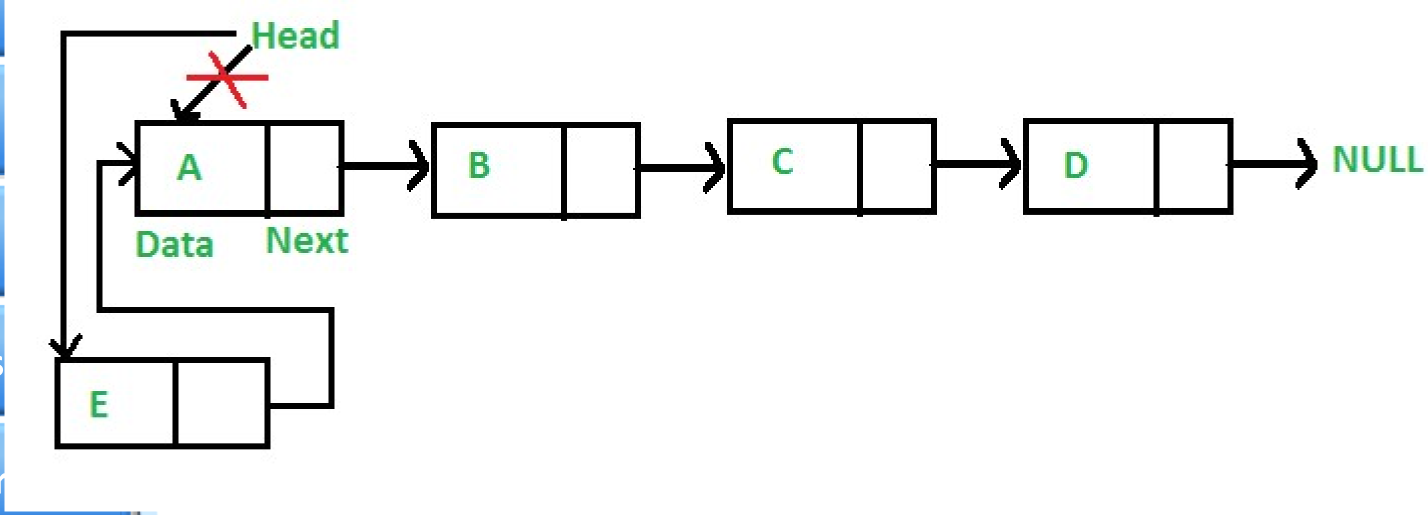
- Create a link list of 5 characters called C.



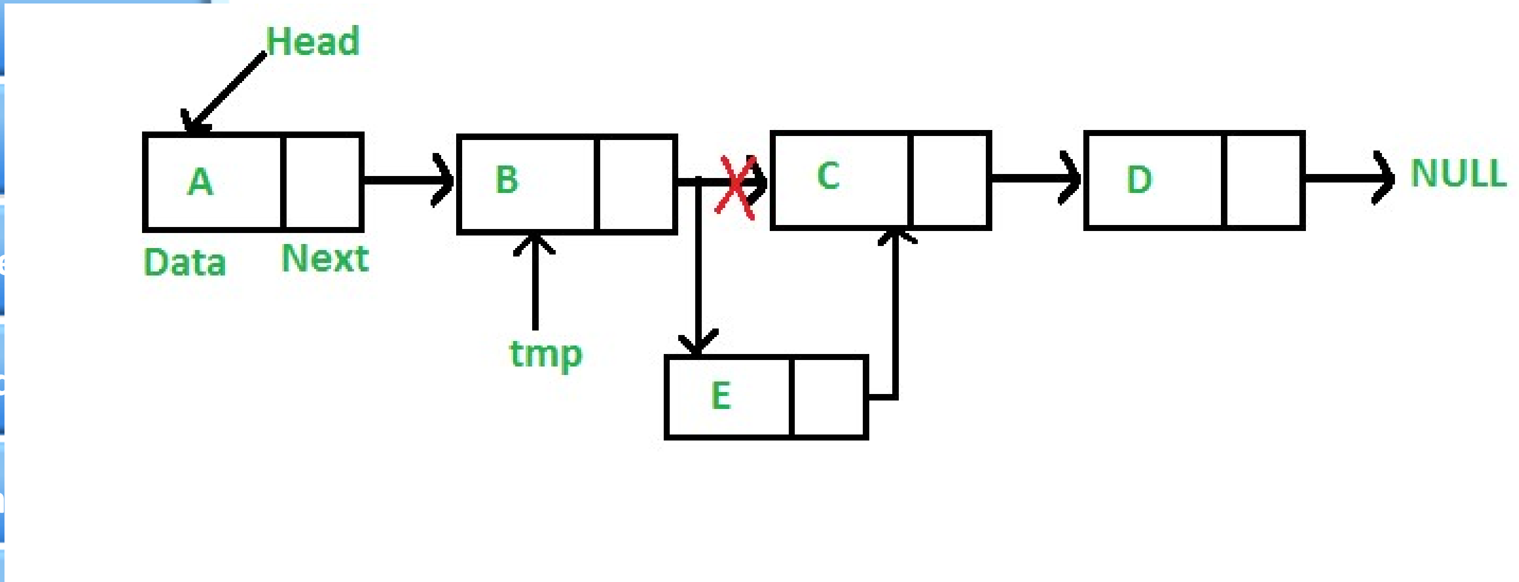
Unit 4: Data Structure

Introduction
Types of DT
Array
Stack
Queue
Linked Lists
Tree and Gr
Hashing
Search & Sort

- Insertion in a linked list : At beginning



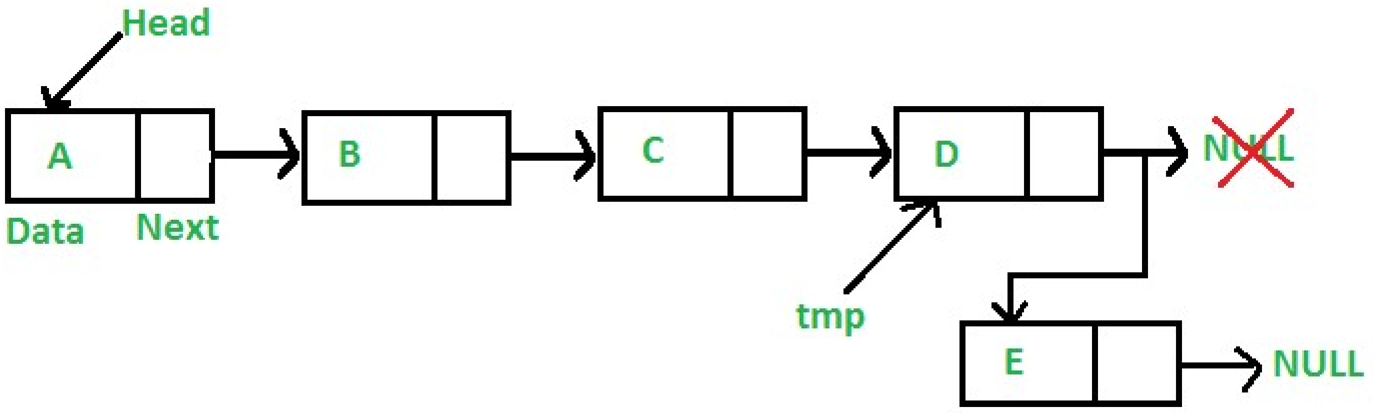
- Insertion in a linked list : In between



Unit 4: Data Structure

- Introduction
- Types of DT
- Array
- Stack
- Queue
- Linked Lis
- Tree and Graph
- Hashing
- Search & Sort

- Insertion in a linked list : At end



Unit 4: Data Structure

Introduction

Types of DT

Array

Stack

Queue

Linked List

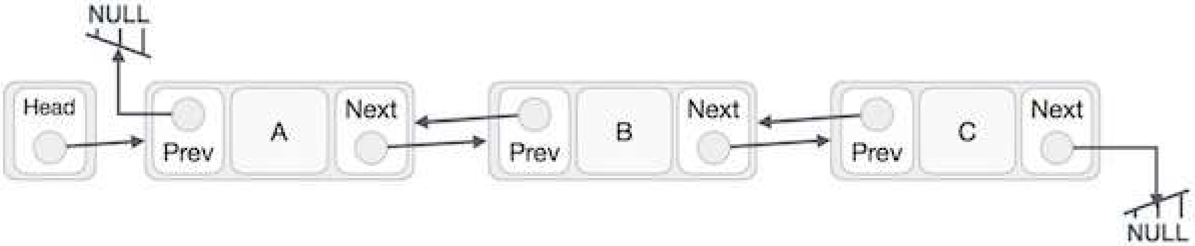
Tree and Graph

Hashing

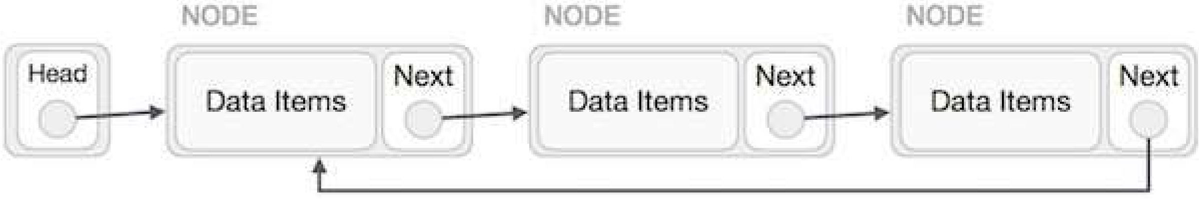
Search & Sort

Types of Linked List

- **Simple Linked List** – Item navigation is forward only.
- **Doubly Linked List** – Items can be navigated forward and backward.

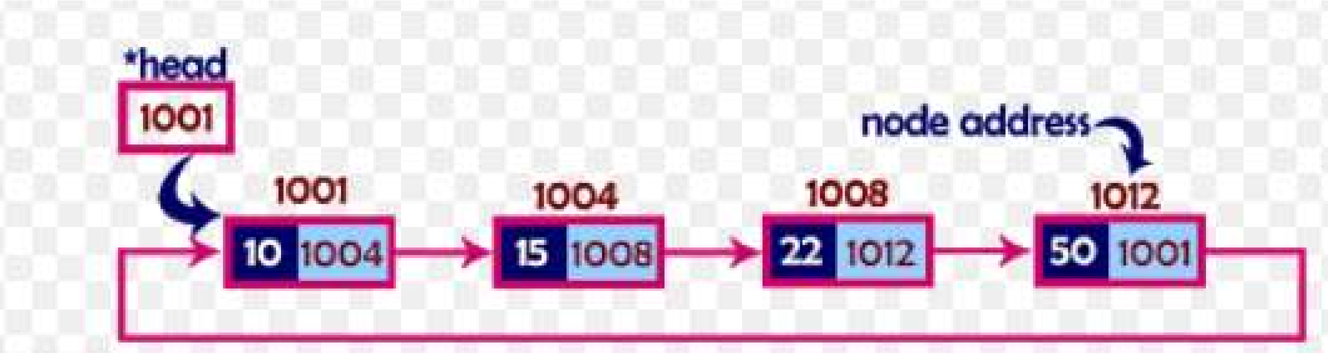


- **Circular Linked List** – Last item contains link of the first element as next and the first element has a link to the last element as previous.



Circular Linked List

- In single linked list, every node points to its next node in the sequence and the last node points NULL. But in circular linked list, every node points to its next node in the sequence but the last node points to the first node in the list.



Java2novice.com

Introduction

Types of DT

Array

Stack

Queue

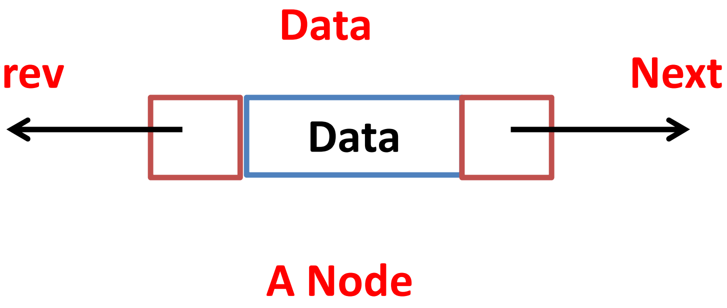
Linked Lists

Tree and Graph

Hashing

Search & Sort

Doubly Linked List Node



Unit 4: Data Structure

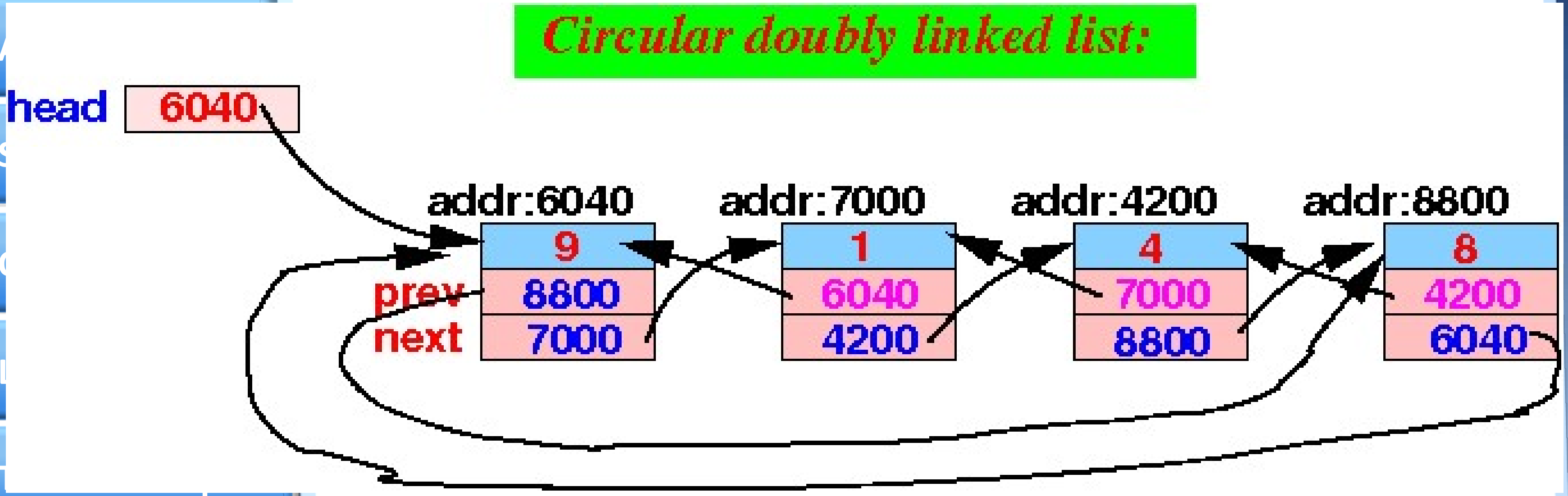
Introduction

Types of DT

Hashing

Search & Sort

Doubly Linked List Example



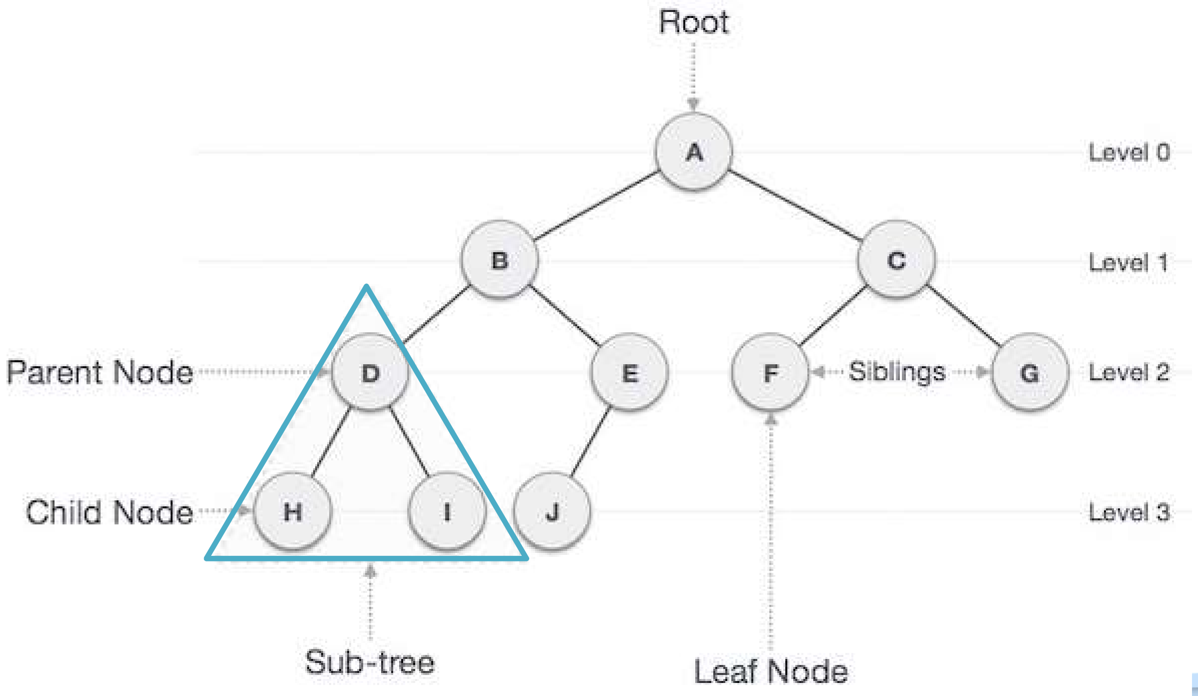
Mathcs.emory.edu

Unit 4: Data Structure

- Introduction
- Types of DT
- Array
- Stack
- Queue
- Linked Lists
- Tree
- Hashing
- Search & Sort

Tree

- **Hierarchical data structure**
- **Tree represents the nodes connected by edges.**
- **If each node is having maximum 2 connected nodes, then it is a binary tree.**



Unit 4: Data Structure

- Introduction
- Types of DT
- Array
- Stack
- Queue
- Linked Lists
- Tree
- Hashing
- Search & Sort

- **A B-tree** is a self-balancing tree data structure that maintains sorted data and allows searches, sequential access, insertions, and deletions in logarithmic time.
- In a **B-tree**, a node can have maximum 'M'('M' is the order of the **tree**) number of child nodes. While in **binary tree**, a node can have maximum two child nodes or sub-**trees**
- **B-tree** is called as sorted **tree** as its nodes are sorted in inorder traversal.
  While **binary tree** is not a sorted **tree**.

Unit 4: Data Structure

- Introduction
- Types of DT
- Array
- Stack
- Queue
- Linked Lists
- **Tree**
- Hashing
- Search & Sort

- B-tree has a height of logM N (Where 'M' is the order of tree and N is the number of nodes).
- While binary tree has a height of log2 N(Where N is the number of nodes).
- B-Tree is performed when the data is loaded in the disk. Binary tree is performed when the data is loaded in the RAM(faster memory).

Unit 4: Data Structure

- Introduction
- Types of DT
- Array
- Stack
- Queue
- Linked Lists
- **Tree**
- Hashing
- Search & Sort

- **A B-tree** is a self-balancing tree data structure that maintains sorted data and allows searches, sequential access, insertions, and deletions in logarithmic time.
- According to Knuth's definition, a B-tree of order *m* is a tree which satisfies the following properties:
- Every node has at most *m* children.
- Every non-leaf node (except root) has at least $[m/2]$ child nodes.
- The root has at least two children if it is not a leaf node.
- A non-leaf node with *k* children contains *k* − 1 keys.
- All leaves appear in the same level and carry no information.

Unit 4: Data Structure

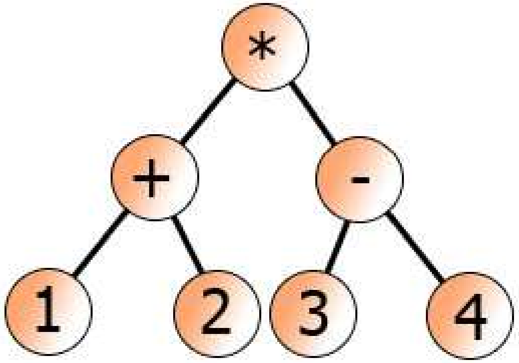
Introduction
Types of DT
Array
Stack
Queue
Linked Lists
Tree
Hashing
Search & Sort

Basic Operations on Tree

- Insertion
- Deletion
- Traversal

Uses of Tree

- Expression handling
- Compilers
- Searching
- Gaming



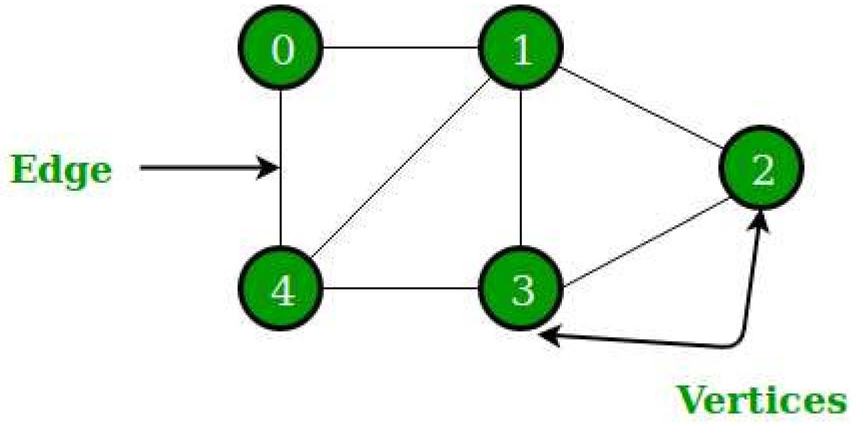
((1+2)*(3-4))

Unit 4: Data Structure

- Introduction
- Types of DT
- Array
- Stack
- Queue
- Linked Lists
- **Graph**
- Hashing
- Search & Sort

Graph

- A **graph** is represented as a set of vertices (nodes or points) connected by edges (arcs or line)
- It is a non linear data structure



Unit 4: Data Structure

Introduction

Types of DT

Array

Stack

Queue

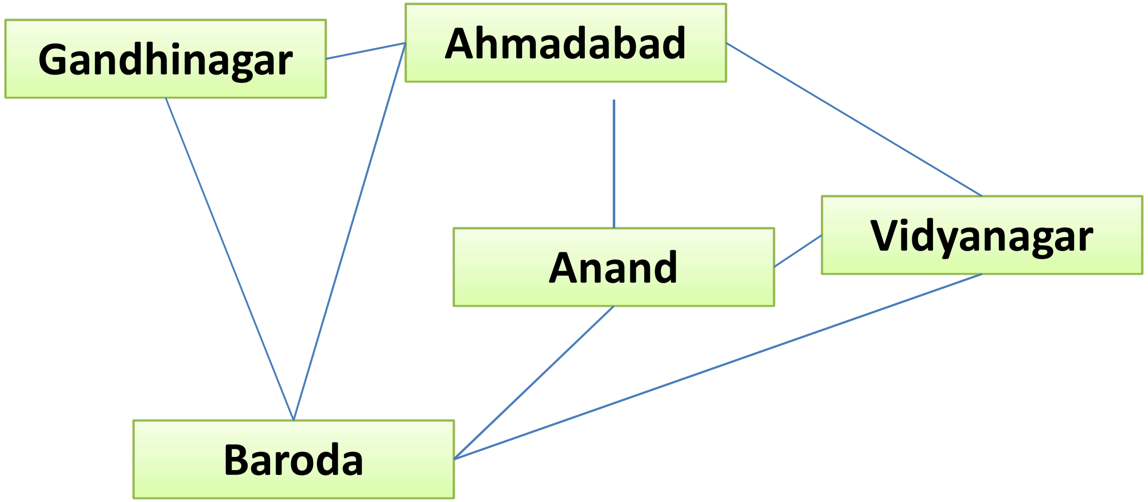
Linked Lists

Graph

Hashing

Search & Sort

Graph Example



Gandhinagar
Ahmadabad
Vidyanagar
Anand
Baroda