Department of Computer Science

Lecture Outline

Data Structures – 4th CSE

Lecture : Linked Lists

Class Lectures on Linked Lists:

Lecture 1 : Introduction to Linked Lists :
https://youtu.be/u3H20LlaL2Y

Lecture 2 : Creation of a Linked List Node:
https://youtu.be/YqXohOxBr6M

Lecture 3: Add at Beginning and End of Linked List:
https://youtu.be/XDb2HXAriG4

Lecture 4: Delete at Beginning and End of a Linked List:
https://www.youtube.com/watch?v=GYzoLbaggdA

Lecture 5: Add/Delete at a Particular Position in a Linked List:
https://www.youtube.com/watch?v=GYzoLbaggdA

Lecture 6: Print, Count and Search in a Linked List:
https://youtu.be/n8eA5-Gw5No

All Programming to be done in C language.

1. Linked Lists	a. What is a list? b. Types of Linked Lists (Singly, Doubly, Circular Linked List) -- Read(https://www.studytonight.com/data-structures/introduction-to-linked-list) c. Advantages of Linked List over Dynamic Arrays d. Comparison of Linked List vs Arrays Read (https://www.geeksforgeeks.org/linked-list-vs-array/) e. Defining the self referential structure for a Singly Linked List f. Creating a single node for a linked list g. Typecasting the allocated data h. Watch (https://www.youtube.com/watch?v=K7VIKlUdo20) i. Implementing Different Functions for Linked Lists i. Create ii. Add Node

	iii. Delete Node iv. Print List v. Count number of elements in the list (Size of List) vi. Add Node at particular position vii. Delete Node at particular position viii. Insert/Delete node at beginning ix. Insert/Delete Node at the end x. Sort elements of Linked List xi. Search Linked List for a particular element xii. Check if List is Empty xiii. Find Nth node of a Linked List xiv. Find some examples here (https://www.edureka.co/blog/linked-list-in-c/) xv. Additional Resource (https://www.learn-c.org/en/Linked_lists) j. Learning Implementation of Linked List by: i. Declaring start pointer to Linked List as a global variable ii. Passing start pointer of Linked List to every function either by value or by reference
2. Doubly Linked List	a. Defining the self referential structure for a Doubly Linked List b. Replicating the functionb56s created for Singly Linked List for Doubly Linked List
3. Circularly Linked List	a. Defining the self referential structure for a Circular Linked List b. Replicating the functions created for Singly Linked List for Circular Linked List c. Creating Doubly Circular Linked Lists as an exercise
4. Applications of Linked Lists	**a. Implementation of stacks and queues** **b. Implementation of graphs : Adjacency list representation of graphs** c. Dynamic memory allocation where linked list of free blocks is used. d. Maintaining directory of names e. Performing arithmetic operations on long integers **f. *Manipulation of polynomials by storing constants in the node of linked list. Implementing polynomial arithmetic for Two polynomials (Add/Subtract/Multiply in code using Linked Lists. (**https://www.daniweb.com/programming/software-development/code/216987/polynomial-addition-using-linked-list) **Topics a,b to be discussed later in detail.**