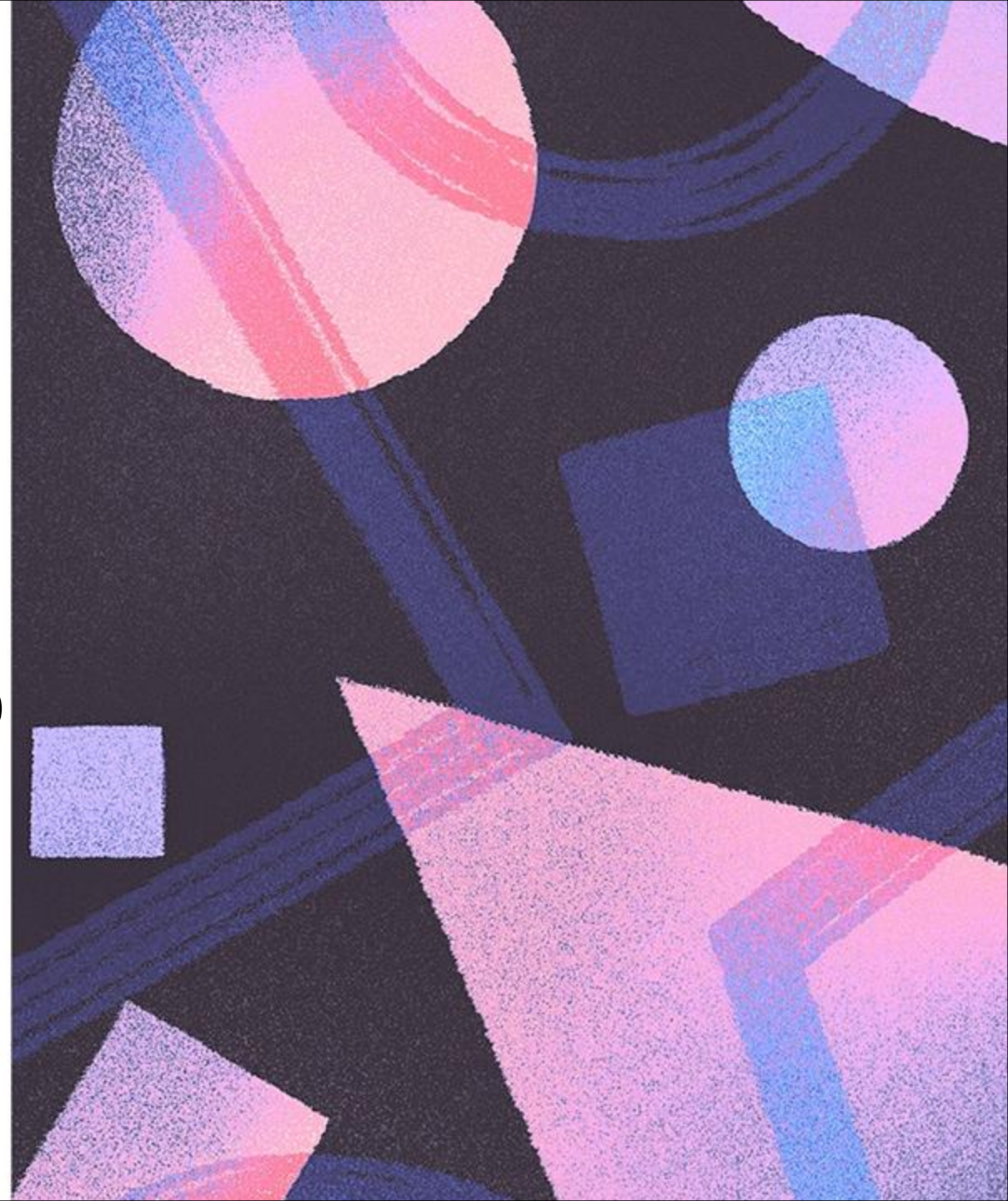


Data Structures Lab

Lecture 2



**Linked Lists – Dynamic
Data Structures**



CONTENT

01

PART

Why Linked Lists ?

02

PART

Single Linked Lists

03

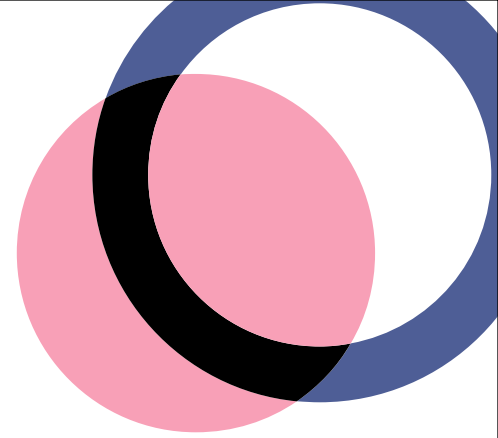
PART

Double Linked Lists

04

PART

Circular Linked Lists



01: Motivation – Why Linked Lists?

- Arrays are common, but they have limitations:
 - Fixed size → need to guess how big the array should be
 - Insertion or deletion in the middle → need to move elements manually
 - Memory must be contiguous → sometimes not enough space

How Linked Lists Solve These Issues

- Linked lists grow dynamically – no fixed size
- Insertion and deletion are easier – no need to move all elements
- Memory does not need to be contiguous

02: Single Linked List (SLL)

- Each node contains:
 - Data → stores value
 - Next → points to the next node
- Last node points to NULL
- Head → stores the first node (or null if the list is empty)

SLL – Basic Operations

1. Insert at head
2. Insert at tail
3. Delete by value
4. Traverse / print list

03: Double Linked List (DLL)

- Each node has:
 - Data
 - Next → points forward
 - Prev → points backward
- Head → stores the first node (or null if the list is empty)
- Tail → stores the last node (or null if the list is empty)

DLL – Operations

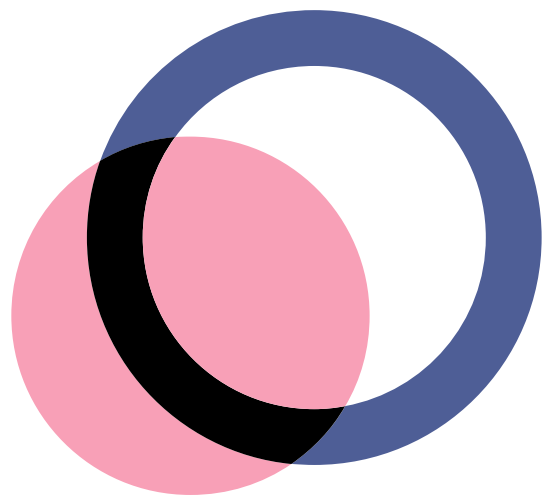
- Insert at head / tail
- Traverse forward / backward
- Delete a node

04: Circular Linked List (CLL)

- Last node points back to head → forms a loop
- Can be single circular or double circular

CLL – Operations

- Insert at end
- Traverse all nodes
- Delete a node



**THANK
YOU**

