



Hospital Emergency Queue System

Doubly Linked List Implementation



Student Name: Ayesha Nawaz Khan



Roll No: 07



Problem Statement

A small hospital needs to manage patients in the Emergency Room (ER) efficiently. Patient priority can change quickly as new critical patients arrive by ambulance or discharged patients leave. The system must support:

- Critical Patient Arrivals: Add patients at the beginning (highest priority)
 - Walk-in Patients: Add patients at the end (normal priority)
- Flexible Positioning: Insert patients at specific positions based on nurse assessment
- Patient Treatment: Remove patients from the beginning after treatment



Proposed Solution

Data Structure: Doubly Linked List

Why Doubly Linked List?

- Efficient insertion at beginning, end, and specific positions
 - Efficient deletion from beginning
 - Bidirectional traversal (forward and backward)
 - Dynamic size, no memory waste
- Each node stores patient ID and pointers to previous and next patients

Graphical Representation of Operations

Initial State

Empty List

HEAD → NULL

TAIL → NULL

Step 01

Insert at End (10)

10

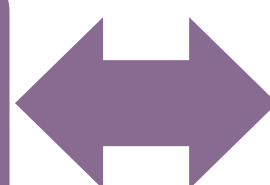
HEAD → 10

TAIL → 10

Step 02

insertBeg(40) //
Critical Patient

40



10

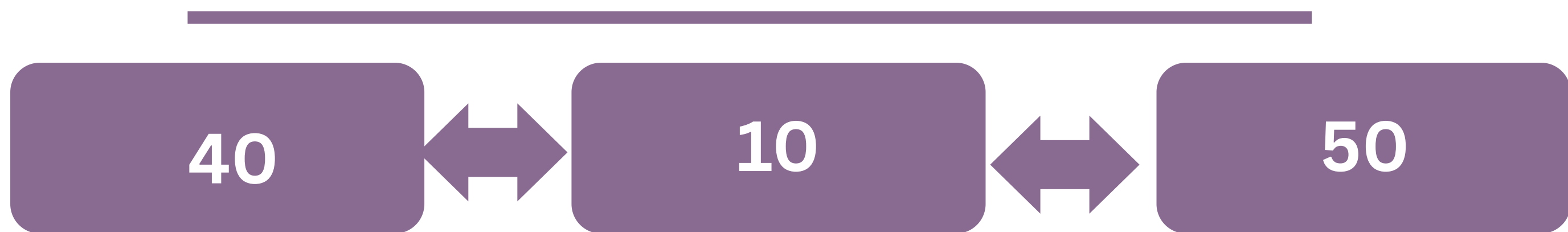
HEAD → 40

TAIL → 10

Graphical Representation of Operations

Step 03

insertAtEnd(50)



HEAD → 40

TAIL → 50

Step 04

insertAtEnd(60)



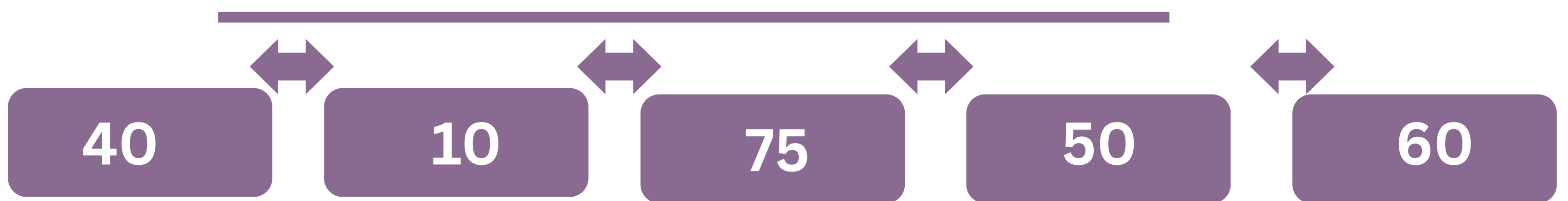
HEAD → 40

TAIL → 60

Graphical Representation of Operations

Step 05

insertAtPos(75, 3)



HEAD → 40

TAIL → 60

Step 06

delFromBeg()



HEAD → 10

TAIL → 60