Hospital Emergency Room Queue System

Student Name: Mughira Ijaz
BS Artificial Intelligence 'B'

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 or treated patients leave. The hospital requires a flexible system to track current patients in the ER using a Doubly Linked List data structure. Each node represents one patient with a patientID and pointers to previous and next patients. The system must support operations like adding critical patients at the beginning, normal patients at the end, inserting at specific positions, and removing treated patients from the beginning.

Proposed Solution

Data Structure: Doubly Linked List with head and tail pointers

Operations Implemented:

• **insertAtBeginning(patientID):** Adds critical ambulance patients who need immediate treatment

• **insertAtEnd(patientID):** Adds walk-in patients who can wait in queue

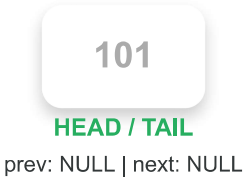
• **insertAtPosition(patientID, position):** Allows head nurse to insert patient at specific position

• **deleteFromBeginning():** Removes treated patient from the front of queue

Edge Cases Handled: Empty list insertion, single node deletion, position out of bounds

Graphical Representation of Operations

Step 1: insertAtEnd(101)



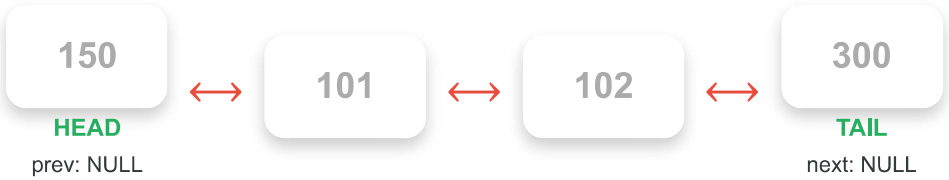
Step 2: insertAtEnd(102)

Step 3: insertAtBeginning(200) // critical patient

Step 4: insertAtPosition(150, 2)

Step 5: deleteFromBeginning()

Step 6: insertAtEnd(300)



Final State After Step 6:

(a) PatientID at head: 150

(b) PatientID at tail: 300

(c) Forward traversal (head → tail): 150 → 101 → 102 → 300

(d) Backward traversal (tail → head): 300 → 102 → 101 → 150