

Doubly Linked List

Advanced Patient Queue Management | Emergency Ward System

PROBLEM STATEMENT

Emergency ward patient queuing inefficiency creates **critical delays** in treatment delivery. Traditional static queue systems fail to dynamically handle **priority insertions** (emergency cases) and standard removals without expensive data shuffling, directly compromising patient outcomes and operational efficiency.

SOLUTION ARCHITECTURE

Doubly Linked List (DLL) implementation provides **$O(1)$ priority insertions**, efficient bidirectional traversal, and **$O(1)$ front/back operations**—completely eliminating array-based data shifting overhead in critical emergency scenarios.

CORE OPERATIONS

Priority Insert

 $O(1)$

Std Insert

 $O(1)$

Delete

 $O(1)$

NODE IMPLEMENTATION

```
class Node {  
  data: PatientID  
  next: Node*  
  prev: Node*  
}
```

KEY ADVANTAGES

✓ Bidirectional Access

✓ $O(1)$ Operations

✓ No Data Shifting

✓ Dynamic Memory

✓ Flexible Insertion

✓ Dual Pointers

1. INSERT(101) → INSERT(102)



2. PRIORITY_INSERT(200) - CRITICAL CASE



3. INSERT_AT_POSITION(150, IDX=2)



4. DELETE_FROM_FRONT() - PATIENT TREATED



Time Complexity Analysis

Front Insertion: $O(1)$ Position Insert: $O(n)$ Bidirectional Traversal: $O(n)$