**ASSIGNMENT**

**NAME: K.SANJAI**

**REG NO: 192325139**

**DEPARTMENT: AI&ML**

**Hospital Bed Allocation**

**Objective:** To allocate hospital beds based on patient priority and bed availability, aiming to minimize wait times and optimize bed utilization.

**Assumptions:**

1. **Patients have priorities** (e.g., critical, high, medium, low).
2. **Bed availability and types** (e.g., ICU, general ward) vary.
3. **Constraints** may include factors like bed type required, special care needs, and department location.

**Step 1: Define Inputs**

1. **Patient Data**:
   * Priority (e.g., critical, high, medium, low).
   * Required bed type (e.g., ICU, general ward).
   * Arrival time.
   * Estimated stay duration.
2. **Bed Data**:
   * Bed type (e.g., ICU, general).
   * Availability status (occupied or free).
   * Department location.

**Step 2: Define Constraints**

1. Critical patients should be prioritized for ICU beds if available.
2. High-priority patients are next in line for specialized beds, followed by medium and low-priority.
3. Bed utilization should be maximized, ensuring minimum idle time.

**Pseudocode for Bed Allocation**

Algorithm HospitalBedAllocation

Inputs: List of patients, List of beds

Outputs: Allocation of beds to patients

1. Sort patients by priority (critical > high > medium > low) and arrival time.

2. Sort available beds by type and department.

3. For each patient in sorted order:

a. Find a suitable bed type and location based on patient’s need.

b. If a matching bed is found:

i. Assign the bed to the patient.

ii. Mark bed as occupied.

iii. Estimate release time based on patient’s expected stay duration.

c. If no bed is available:

i. Place patient in a waitlist with priority and expected wait time.

ii. Check periodically for bed release and reassign.

4. Update bed occupancy data.

5. Calculate and display wait time reduction and occupancy percentage.

End Algorithm

**Complexity Analysis**

* **Time Complexity**:
  + Sorting patients: O(nlogn)*O*(*n*log*n*), where n is the number of patients.
  + Sorting beds: O(mlogm)*O*(*m*log*m*), where m*m* is the number of beds.
  + Finding a matching bed for each patient in the worst case: O(n×m)*O*(*n*×*m*).

**Overall**: O(nlogn+mlogm+n×m)*O*(*n*log*n*+*m*log*m*+*n*×*m*).

* **Space Complexity**:
  + Storing patients and beds requires O(n+m)*O*(*n*+*m*) space.

**Simulation & Analysis**

**Simulation Parameters**:

1. Set a random list of patients with varying priorities and arrival times.
2. Set a fixed number of beds (some occupied, some free).

**Expected Outcome**:

* Reduced wait time for high-priority patients.
* High bed occupancy, minimizing idle time.
* A dynamic waitlist that updates as beds are released.

**Conclusion**

In conclusion, this hospital bed allocation algorithm effectively minimizes patient wait times and maximizes bed utilization by prioritizing patients based on urgency and dynamically managing bed availability. By incorporating a waitlist system and periodically updating bed status, the algorithm ensures that resources are allocated efficiently, leading to improved patient care and optimized hospital operations.