Data Structures and Algorithms

HEALTHCARE MANAGEMENT SYSTEM

Course Project Report

School of Computer Science and Engineering

2024-25

Contents

1. Course and Team Details

1.1 Course details

Course Name	Design and Analysis of Algorithms
Course Code	24ECSC205
Semester	III
Division	A
Year	2024-25
Instructor	Mallikarjun Akki

1.2 Team Details

Si. No.	Roll No.	Name
1.	118	Prasad Morab
2.	120	Prasanna Anagal
3.	137	Siddharth Prabhu
4.	140	Shashank Joshi

1.3 Report Owner

Roll No.	Name
140	Shashank Joshi

2. Introduction

The Healthcare Management System (HMS) is a project that uses efficient algorithms and data structures to optimize healthcare operations. In a city, health of citizens matters the most. So this project is picked although there were many problems that may be addressed. HMS addresses four key areas:

Emergency Incident Categorization – categorizing and prioritizing medical emergencies;
Patient Appointment Scheduling – scheduling appointments for patients in an efficient manner;
Medical Supply Inventory Management – tracking and restocking of medical supplies;
Healthcare Resource Allocation – optimizing the critical distribution of healthcare resources.
HMS is done with the help of algorithms of Dijkstra's Algorithm, the Rabin-Karp string matching, QuickSort, Depth First Search, and Priority Queues, BST, and Union Find to bring improved efficiency in timely real-time decision making.

It is based on the principles from algorithmic problem-solving and data structures, discussed in key references like Data Structures and Algorithm Analysis in C++ by Weiss [1] and Introduction to Algorithms by Cormen et al. [2]. This helped choose suitable algorithms for every module, scalable enough for the healthcare scenario, while being very efficient in practice.

3. Problem Statement

3.1 Domain

The HMS automates healthcare operations such as incident management, patient scheduling, resource allocation and inventory management. These tasks are full of inefficiencies and delays, leading to poor service quality and increased costs. The system's objective is to resolve these challenges by providing an algorithm-driven, scalable, and user-friendly solution.

3.2 Module Description

The **Healthcare Resource Allocation System** ensures that critical healthcare resources such as hospital beds, medical staff, and life-saving equipment are distributed optimally. The system employs **graph-based pathfinding, disjoint-set management, and priority-based allocation** for efficiency.

Key Algorithms Used:

1. **Dijkstra's Algorithm** – Finds the shortest path for resource transportation.
2. **Union-Find** – Groups related resources for better management.
3. **Priority Queue** – Allocates resources based on urgency..

4. Functionality Selection

Si. No.	Functionality Name	Known	Unknown	Principles applicable	Algorithms	Data Structures
	Name the functionality within the module	What information do you already know about the module? What kind of data you already have? How much of process information is known?	What are the pain points? What information needs to be explored and understood? What are challenges?	What are the supporting principles and design techniques?	List all the algorithms you will use	What are the supporting data structures?
1	Optimizing Resource Transport	Resources (e.g., ambulances, ventilators) must be transported quickly between hospitals.	Finding the most efficient route dynamically.	Shortest Path	Dijkstra's Algorithm	Graphs
2	Grouping Related Resources	Some resources (e.g., ICU beds + ventilators) must be allocated together.	Efficiently managing grouped resources without redundancy.	Disjoint Sets	Union-Find	Arrays
3	Priority-Based Allocation	Some resources are more urgently needed than others.	Allocating resources fairly based on priority.	Greedy Approach	Priority Queue	Heaps

5. Functionality Analysis

5.1 Optimizing Resource Transport (Dijkstra's Algorithm)

- **Workflow:**
  1. A hospital requests a resource (e.g., ventilator).
  2. Dijkstra's Algorithm calculates the shortest path to transport it.
  3. The most efficient route is chosen to minimize response time.
- **Efficiency:**
  - **Time Complexity:** O(V^2) for adjacency matrix, O((V+E)logV) for priority queue-based implementation.
  - **Space Complexity:** O(V^2) for matrix, O(V+E) for adjacency list.

5.2 Grouping Related Resources (Union-Find)

- **Workflow:**
    1. Resources that must be allocated together (e.g., medical staff + equipment) are grouped.
    2. Union-Find merges resource sets dynamically.
    3. Efficiently determines if a resource belongs to an existing group.
- **Efficiency:**
    - **Time Complexity:** O(α(n)) per operation, where α(n) is the inverse Ackermann function (near constant).
    - **Space Complexity**: O(n)

5.3 Priority-Based Allocation (Priority Queue)

- **Workflow:**
    1. Requests are inserted into a priority queue based on urgency.
    2. The most urgent request is processed first.
    3. Ensures fair and efficient resource distribution.
- **Efficiency:**
    - **Time Complexity:** O(log n) for insertions and deletions.
    - **Space Complexity:** O(n) for maintaining the heap

6. Conclusion

The Healthcare Resource Allocation System ensures optimal distribution of medical resources such as hospital beds, staff, and critical equipment. Using Dijkstra's Algorithm for shortest path calculations, Union-Find for efficient resource grouping, and Priority Queues for urgent allocation, the system enhances decision-making in resource distribution.

This module demonstrates how graph-based algorithms can be leveraged to optimize real-time logistics in healthcare. Efficient resource allocation reduces delays, minimizes wastage, and improves response times in critical situations. By ensuring that urgent needs are met first, this system contributes to the overall effectiveness of the healthcare sector.

7. References

[1] **Weiss, Mark Allen.** *Data Structures and Algorithm Analysis in C++ (4th Edition).* Pearson, 2013.

[2] **Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson.** *Introduction to Algorithms (3rd ed.).* The MIT Press, 2009.

~*~*~*~*~*~*~*~