# Task 4 Report – Integrated Hospital System

Task 4 is the most advanced part of the assignment. In this task, all previously implemented data structures are combined into a single Hospital Management System. The system uses Linked List, Queue, Stack, HashMap and Sorting concepts together.

The PatientList (Linked List) is used to store admitted patients. The TreatmentQueue manages treatment requests, and the DischargeStack stores discharged patients. A HashMap is also used to store patient information with patient id as key. This allows fast patient lookup in **O(1)** average time.

For priority treatment handling, two separate queues are used: one for priority patients and one for normal patients. Priority patients are always processed first, but FIFO order is still preserved inside each queue. The enqueue and dequeue operations still run in **O(1)** time. This approach is simple and efficient.

A sorting algorithm can be used to sort patients by severity level or waiting time. For example, Bubble Sort has $\mathbf{O(n^2)}$ time complexity and is easy to implement, but it is not efficient for large data. Merge Sort has **O(n log n)** complexity and performs better for bigger systems.

If a heap-based priority queue was used instead, the system could handle priorities more efficiently. Insert and remove operations would work in **O(log n)** time and priority handling would be automatic. This would improve performance especially when the number of patients increases.

Overall, this integrated system shows how different data structures can work together to solve a real-world hospital management problem.