

# EECS 560 - Lab 7 Report

---

**Author: Jace Kline 2881618**

## Answers to Questions

---

1. Explain why it is advantageous to use priority heaps in hospital management systems.

Priority heaps are very useful to utilize in a hospital management system because they offer a system/mechanism for very quickly and efficiently getting information/data based on a defined priority measure (based on the particular application). In the context of a hospital, this can be used to track records of patients where the priority key is an algorithm that takes many factors like age, overall health, severity of injury to compute a key which will essentially "rank" the patients in terms of who needs to be seen the quickest. In addition to keeping track of the patients, a priority heap could also track the doctors, nurses, or any other shared (or in demand) resource at the hospital. Since hospitals are complicated and dynamic systems with multiple moving parts, employing a priority-based algorithmic approach to managing resources/people can produce efficiency, productivity, and help cut down on confusion and chaos.

2. Explain a few features which should be incorporated into the hospital patient and doctor management system for the system to be useful in an actual (i.e., real-world hospital) for managing emergency room patient cases.

A few features that should be implemented into the hospital patient and doctor management system are:

- An algorithm that takes various components of data about a patient (like overall health, recent health issues, severity of the current issue, current issue type, etc.) and automatically computes a priority value so that the patient can be quickly added into the management system without manual assignment of priorities.
- A way to "weight" doctors more heavily to certain emergency room cases than others, based on the doctor's speciality and skillset. This would, of course, have to be only one of many factors to be taken into account when scheduling doctors.
- For the operators/users of this management program, it would be ideal to implement a more user-friendly graphical user interface.
- Multiple priority heaps (redundancy) for each set of resources/people where each heap's priority value measures a different factor about the current situation. This preserves more information than simply combining lots of variables (factors) to reduce to a single priority value. However, this approach would increase complexity and memory cost.

3. If we need to apply priority heaps in product recommendations (based on features or product ratings), how can we apply them?

Using priority heaps in the context of features or ratings would be quite straightforward. You would simply have to find or define the algorithm/function that computes a priority key (for given types of items) based on whatever criteria about those items one is interested in. The heap objects would be cyber-representation records of some objects and the priority keys associated with these record objects would be used to capture the relational/comparative component between the objects (based on the criteria one is interested in ranking). For example, a priority heap that tracks objects and their ratings would simply use the rating as the priority key in a max heap.