Analysis Questions and Answers for Task 4

* Explain the Big O complexity of your priority and sorting mechanisms.

-) To control patient flow, the current system makes use of two straightforward linear queues (TreatmentQueue) and a DoublyLinkedList (PatientList).

Sorting mechanism's time complexity equals O(n^2)

A Bubble Sort algorithm is implemented by PatientList's sortBySeverity() method. It repeatedly goes through the list comparing neighboring nodes and switching their data if the severity of the current patient is less than that of the subsequent patient. Bubble Sort has an O(n^2) time complexity in the worst and average scenarios because it needs two nested loops to make sure the list is sorted.

Priority Mechanism's time complexity equals to O(1) for enqueueing and O(n) for processing.

Currently, HospiatalSystem handles the priority system by dividing requests into two distinct queues (normalTreatmentQueue and priorityTreatmentQueue).
Adding a request only adds to the end of a linked list queue, so it is O(1).
Every treated patient has O(n) complexity since the PatientList is a linkedlist and the removal is based on an Id search (a linear scan).

* How would using a heap-based priority queue improve performance?

-) In the current implementation, we need to either sort the entire list 0(n^2) or conduct a linear search O(n) to find the most severe patient. The patient with the greatest severity always kept at the base of the tree when using a heap.

The heap bubbles a new patient up to their proper position in logarithmic time O(logn) which is much quicker than sorting. The system extracts the root when a doctor is prepared to see a patient. After that, the heap rearranges itself in O(logn) time. An O(n2) operation could require 1000000 steps for a hospital with 1000 patients, while an O(logn) operation would require about 10 steps. This change guarantees that the system will continue to function even during times of high demands.

Output of The Test Scenario

Added: Sponge Bob
Added: Patrick Star
Added: Mr Crabs
Added: Ryan Gosling

Added: John Cena
Added: Hüseyin Alp
Added: John Doe
Added: Lorem Ipsum
Added: Memento Mori
Added: Veni Vidi Vici
Removed patient Name: Lorem Ipsum ID: 108
Removed patient Name: Memento Mori ID: 109
Removed patient Name: Veni Vidi Vici ID: 110
Removed patient Name: Sponge Bob ID: 101

*** Patients Sorted by Severity ***

*** Current Patient List ***
* Patient's ID: 105, Name: John Cena, Severity: 10, Age: 40 *
* Patient's ID: 106, Name: Hüseyin Alp, Severity: 10, Age: 20 *
* Patient's ID: 102, Name: Patrick Star, Severity: 6, Age: 25 *
* Patient's ID: 104, Name: Ryan Gosling, Severity: 6, Age: 32 *
* Patient's ID: 107, Name: John Doe, Severity: 6, Age: 22 *
* Patient's ID: 103, Name: Mr Crabs, Severity: 4, Age: 30 *
****

*** Current Patient List ***
* Patient's ID: 105, Name: John Cena, Severity: 10, Age: 40 *
* Patient's ID: 106, Name: Hüseyin Alp, Severity: 10, Age: 20 *
* Patient's ID: 102, Name: Patrick Star, Severity: 6, Age: 25 *
* Patient's ID: 104, Name: Ryan Gosling, Severity: 6, Age: 32 *
* Patient's ID: 107, Name: John Doe, Severity: 6, Age: 22 *
* Patient's ID: 103, Name: Mr Crabs, Severity: 4, Age: 30 *
****
*** Pending Priority Requests ***
0
*** Pending Normal Requests ***
4
*** Discharge History ***
Records (Top > Bottom): * Patient's ID: 101, Discharge Time: 800 * ->
* Patient's ID: 110, Discharge Time: 1500 * ->
* Patient's ID: 109, Discharge Time: 1400 * ->
* Patient's ID: 108, Discharge Time: 1300 * ->
* Patient's ID: 107, Discharge Time: 1700 * ->
* Patient's ID: 106, Discharge Time: 1600 *