COMP 352 Data Structures and Algorithm
Fall 2020: Assignment 3 Programming

Name: Jordan Jeffrey Chan Kum Sang
Student ID: 40125997
Professor: Nora Houari
Section: Section FF
Date: November 14,2020

Part III

1. Sorted List Priority Queue

The Big-0 time complexity of the unsorted list is O (1) for insertion, O(n) for deletion since you need to search the list.
The Big-Omega complexity of the unsorted list is O (1) for insert still, and O(n) for deletion since we still have to iterate through the list no matter what.

2. Unsorted List Priority Queue

The Big-O time complexity of the sorted list is O(n) for deletion, but O(n) for insertion since it needs to find where to be placed.
The Big-Omega complexity of the sorted list is O(n) for deletion/insertion as well since even in the best case you have to shift the array if you insert, and you still have to sort if you delete.

3. Array-Based Heap Priority Queue

The Big-O and Big-Theta time complexity for the array-based heap is O (nlog n) for insertion and removals.
The Big-omega complexity of the array-based heap is O (log n) for insertion and removals as well.

The Space complexity of the sorted/unsorted list is O(n) since the main space requirement is to build the list.
The Space complexity of the sorted list is O(n) since the main space requirement is to build the heap.

Yes. There is a noticeable performance difference between mostly the unsorted/sorted list implementation versus the heap implementation. The heap implementation is incredible fast, and this is especially noticeable once we go past n = 10,00 – it is roughly 3000% faster than the sorted or unsorted.

In terms of sorted versus unsorted list the difference isn't too noticeable as they should both share the same time complexity.