**CS 232: DATA STRUCTURE – FINAL PROJECT**

**QUESTIONS FOR C LEVEL**

By Kieu My Vo

1. **Which version of the list-based Priority Queue did you implement?**

**Why did you pick that one?**

The version of the list-based Priority Queue is *“A PQ where the enqueue method always adds an item to the end of the list, and the dequeue method searches the list for the highest-priority item and removes it from wherever it is”.*

*(If you remove a node from the middle or end of the list you will need to make sure to adjust the pointers to the nodes around it so the list doesn’t break)*

I chose the first version since it is simpler to implement since new nodes are always added to the end of the list. Moreover, the first version may be more efficient for a small list. In contrast, the second version searching for the correct insertion point can be slow if the list is very long. This is because the new item should be inserted in order, which means we have to re-arrange the list every time a new item is added.

1. **What are the Big O times for the enqueue, dequeue, size, is\_empty, and \_\_str\_\_ functions for the PQ you implemented? Explain your answers**

* \_\_str\_\_ is O(n) because it has to print everything in the queue, and we don't know how long that will be.
* The dequeue() method is O(n) because it uses a loop to find the highest-priority item, which will go through all items in the list and we don’t know how long that will be. Moreover, this is also because in the case when removing the highest\_priority item which is at position 0, it modifies the list at position 0, which means that every time something is removed the entire rest of the list has to move as well.
* The enqueue() method is O(1) because it modifies (adds item) the end of the list, and changes to that end don't affect anything else.
* \_\_init\_\_ is O(1) since this function creates an empty LinkedQueue and does not relate to the growth rate.
* size() is O(1 because it returns the value of the length of the LinkedQueue and doesn't affect anything else.
* is\_empty is O(1) since it simply checks whether the Queue is empty or not, which takes a constant amount of time.