Thomas Rafeld, M12808058

Ryan Winterhalter, M12727389

Lab 7 Report

**Objectives:**

The objectives for this assignment were to be able to describe a Priority Queue and to be able to describe and understand how to build a Heap. The first objective is important to this class for two reasons. The first reason is because the first task of this lab is to create a priority queue, so understanding and being able to describe one is essential to understanding it. The other reason is because it is an important data structure, which is what this class is about, so we need to be able to describe them and how they work to fully understand data structures. This is needed for a career in engineering as, for a project, we may be working on one where we might need to make a priority queue, so being able to describe and make one would be essential. The second objective is important to the class for a similar reason. It is needed in the lab specifically as for task 2 we had to make a Max Heap, so being able to describe a heap and understanding how to build one is essential to that task. Also, as it is a data structure, we need to understand it for this class as a whole, as the overall objective is to learn about and understand data structures. This objective is important to a career in Engineering as we may need to implement a heap or teach someone how to use one, so being able to describe a heap and understanding how to build one would be important for that.

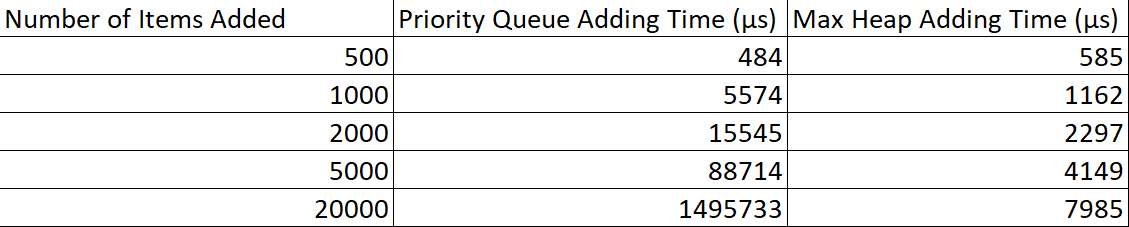
**Task 1:**

An advantage of the priority queue is that it is sorted by priority so the most important thing is always the first removed. Therefore, not only can the user sort items based on a set priority, but removing items from a sorted list priority queue is much quicker and more efficient because there isn’t any shifting of values within the queue, as the sorted linked list will use nodes to remove the item from the priority queue. A disadvantage would be that it takes a lot longer to add to a priority queue since it has to search through the queue to find where the item goes, whereas a normal queue already knows where to insert the next item in the data from its array implementation.

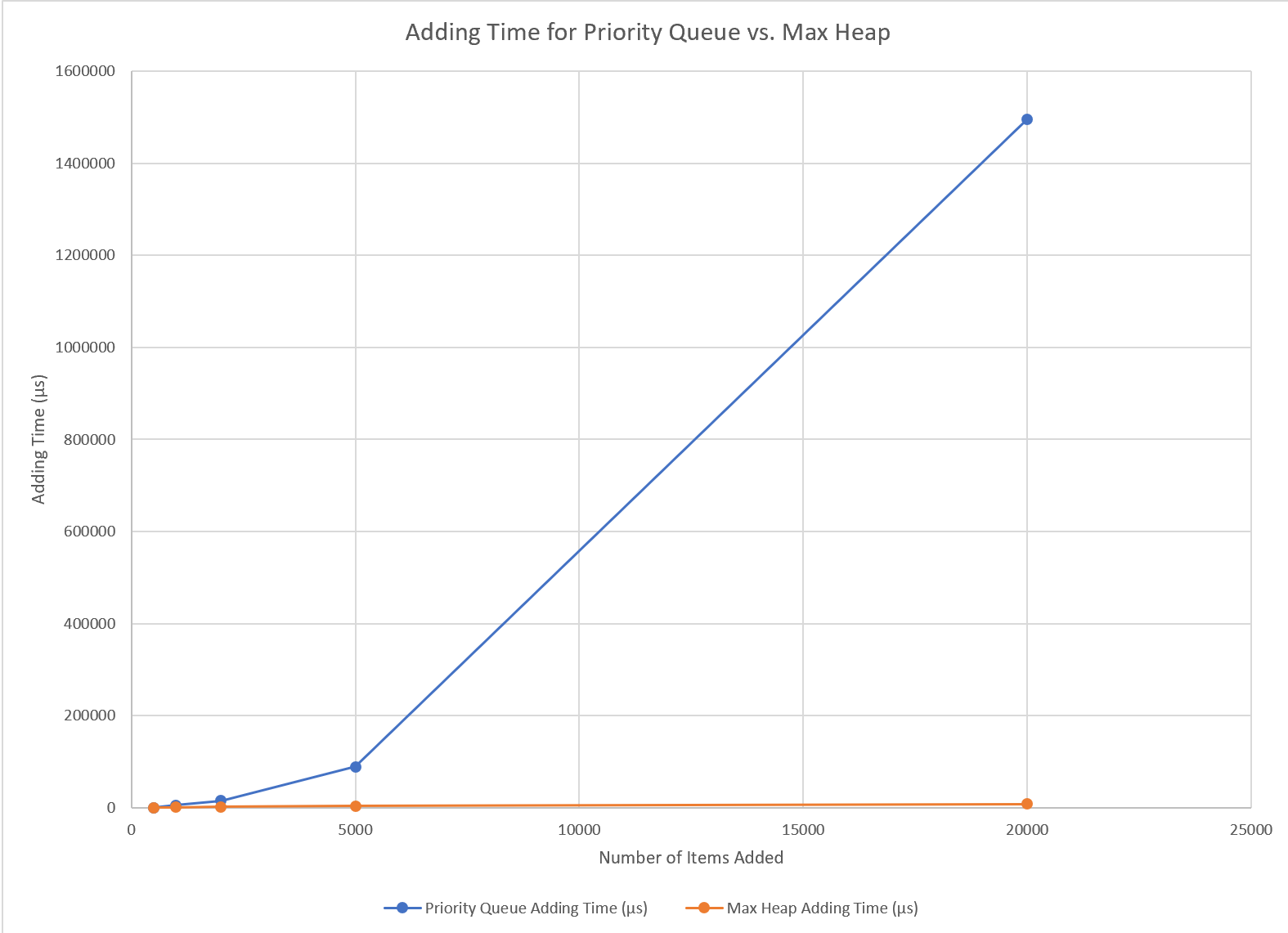
**Task 2:**

One advantage of a max heap compared to the binary search tree is that the items are stored in an array so memory leaks shouldn’t happen (compiler will automatically handle destruction of the array). Another advantage would be that the heap is sorted by max priority while a binary search tree is not already sorted so that has to be added to the binary search tree. The remove time in a heap is faster as it just removes the root, which is the max value depending on the heap implementation, so this would be another advantage. A disadvantage would be that in the heap, you can only remove the root while in the binary search tree you remove whatever value you want. Another disadvantage is that the find time for the max heap would be much greater because the left and right child isn’t sorted by a certain standard while it would be less in the binary search tree.

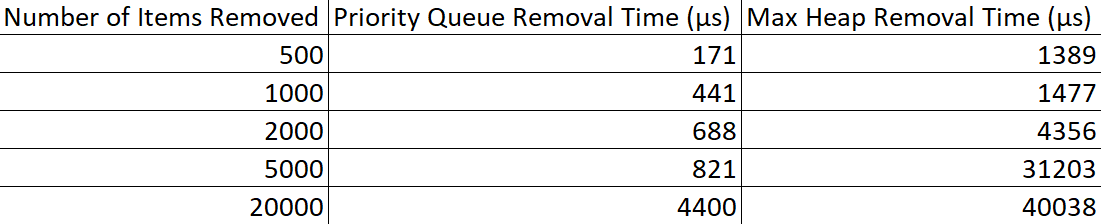
**Task 3:**

****

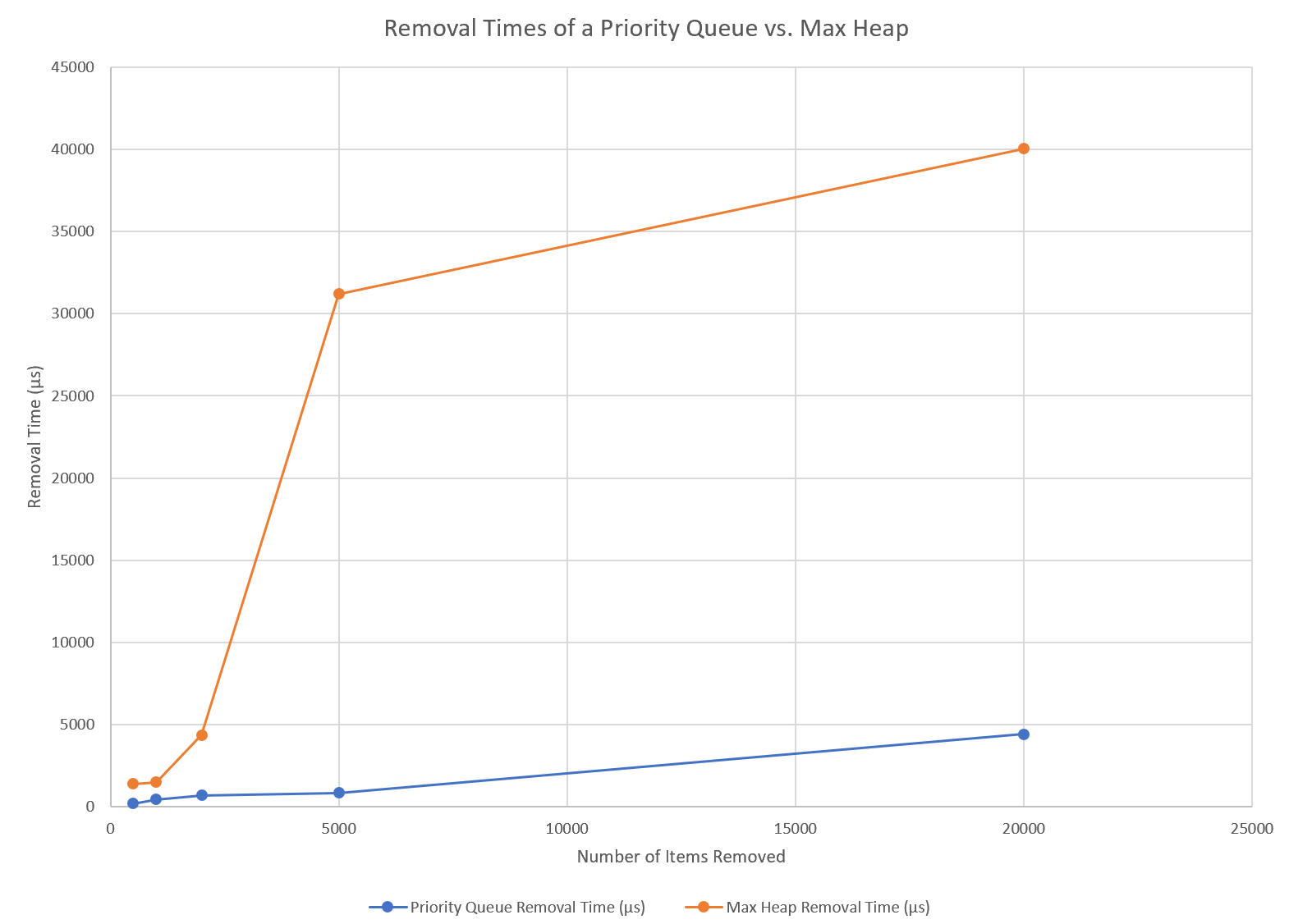
The above table shows the performance results of our priority queue and max heap for adding items to the data structures.

****

The structures each performed based on our expectations. Adding items to a priority queue can take more time, especially as more items are added and the entire list needs to be searched, whereas with a max heap items only need to be added and followed by, at most, a few swaps. Therefore, we expected our priority queue to have a significantly greater adding time than our max heap, especially as the number of items being added to the structures increased.

****

The above table shows the performance results of our priority queue and max heap for removing items from the data structure.

****

The structures each performed based on our expectations. Removing items from a sorted linked list is quite simple and quick since only the first item from the list needs to be removed, whereas with a max heap, removing an item may require multiple swaps to occur. Therefore, we expect our max heap to have significantly greater removal times versus the priority queue, especially as the number of items being removed from the structures increased.

**Group Contributions:**

The lab was worked on together by both Ryan and Thomas while on a call together in Microsoft Teams. We used Thomas’s header file for task 1 and his ‘.cpp’ files for tasks 1 and 2. We used Ryan’s header file for task 2 and his ‘.cpp’ file for task 3. Ryan gathered the performance data. For the final grade each member of the group should receive 100 percent of the grade as we feel that we both evenly contributed to the lab and worked together for almost the whole time it was being worked on.