Grant Mercer

CS 302

Assignment 9 Part B

**Priority Queue:**

A priority queue is much like a regular queue or stack, but additionally each element has a priority associated with it. In a priority queue the element with the highest priority is served before the elements with a lower priority. This can be useful in situations such a printer queues where certain documents may have a higher priority to be printed over others.

**Priority Queue vs Balanced binary search tree**

A balanced binary search tree can guarantee the user that all elements are ordered from left to right, while a priority Queue guarantees that the elements will be greater/smaller than lower elements. A large advantage of a Priority Queue is that the run time complexity of findMin / findMax is O(1), and the BBST is O(logN) for all finds. In an application where the Min or Max is accessed often the Priority Queue would be a much better option to prevent a slowdown by O(logN) with a BBST.

**Consquences of using buildHeap and insert incorrectly**

Calling insert and buildHeap can have some large consequences in the program. It is important to remember the distention between *buildHeap* and *insert*. *buildHeap* will set up the data to follow the structure and order properly, while *insert* places a **single** element into the heap and makes the appropriate adjustments to maintain the order properly. This means insert only makes sure the order in maintained with respect to the single element inserted, but buildHeap makes sure all data follows the order and structure property.

**Big-O for the various operations**

Insert – O(log n)

BuildHeap – O(N)

DeleteMax- O(log N)

ReheapUp- O(log N)

ReheapDown- O(log N)

Resize- O(n)