

### **Assignment 3**

#### **Method Time Complexities**

##### **Toggle():**

It will be  $O(n)$  since we have to traverse the entire queue to sort them out.

##### **Remove(e):**

My implementation needs to traverse the queue to check if the object exists there first, then it removes the entry and returns it. So it would be  $O(n)$  in the worst case.

##### **replaceKey(e, k):**

First we need to check if the entry exists in the queue, then we replace the key of that entry with the new key. Then we need to restore the order of the heap, which may require  $O(\log n)$  time. So the overall time complexity would be  $O(n \log n)$

##### **replaceValue(e,v):**

This one will have  $O(n)$  time complexity only, because we only check if the entry is in the queue or not. Then we change the value of the entry and that does not require restoring the order of the queue.