Kristine Trinh
11190412
nlt895

Question 1)

Name: PriorityQueue<G>
Set:
Q: set of queues containing items from G
G: set of items that can be in the queue
B: {true, false}
N: set of positive integers
N_0: set of non-negative integers

Signatures:

newPriorityQueue<G>: $N \to Q$

Q.insert(g): $G \nrightarrow Q$

Q.isEmpty: $\to B$

Q.isFull: $\to B$

Q.maxItem: $\nrightarrow G$

Q.minItem: $\nrightarrow G$

Q.deleteMax: $\nrightarrow Q$

Q.deleteAllMax: $\nrightarrow Q$

Q.deleteMin: $\nrightarrow Q$

Q.frequency(g): $\to N\_0$

Preconditions: For all q∈Q, g∈G, n∈N_0

newPriorityQueue<G>(n): none
q.insert(g): q is not full
q.isEmpty: none
q.isFull: none
q.maxItem: q is not empty
q.minItem: q is not empty
q.deleteMax: q is not empty
q.deleteAllMax: q is not empty
q.deleteMin: q is not empty
q.frequency(g): none

Semantics: For all q∈Q, g∈G, n∈N_0

newPriorityQueue<G> (n): create a new queue of items G with capacity n
q.insert(g): enqueues an item g in highest priority order
q.isEmpty: returns true if queue is empty, false otherwise
q.isFull: returns true if the queue is full, false otherwise
q.maxItem: returns the item with highest priority in q
q.minItem: returns the item with lowest priority in q
q.deleteMax: deletes the item g with highest priority in q
q.deleteAllMax: deletes every the items with the highest priorities from q
q.deleteMin: deletes the item g with lowest priority in q
q.frequency(g): returns the number of times given item occurs in q

Question 2)

The worst-case time complexity of the heap insertion and deletion will have the same time complexity, since the number of iteration of both methods depends on the height of the heap. While insert() goes from top to bottom, delete() goes in the reverse direction: from bottom to top.

Running time of insertion and deletion into heap of **n** nodes = height of the heap
Let n = number nodes in heap of height h

$2h - 1 < n \leq 2h+1 - 1$

$2h < n + 1 \leq 2h+1$

$h < \log(n + 1) \leq h+1$

Height of heap ~= log (n + 1)
∴The worst-case time complexity for deletion and insertion will be $\boldsymbol{O(logn)}$