# 1 Solutions

## Question 1 ():

**Solution:**

**Name:** PriorityQueue<$G$>

**Sets:** $Q$ : set of priority queues containing elements from $G$.
$G$ : set of items that can be in a priority queue.
$B$ : $\{true, false\}$
$\mathbb{N}$ : set of positive integers.
$\mathbb{N}_0$ : set of non-negative integers.

**Signatures:** newPriorityQueue<$G$>(n): $\mathbb{N} \to Q$
$Q$.insert(g): $G \not\to Q$
$Q$.isFull: $\to B$
$Q$.isEmpty: $\to B$
$Q$.maxItem: $\not\to G$
$Q$.minItem: $\not\to G$
$Q$.deleteMax: $\not\to Q$
$Q$.deleteMin: $\not\to Q$
$Q$.frequency(g): $G \to N_0$
$Q$.deleteAllMax: $\not\to Q$

**Preconditions:** For all $q \in Q$, $g \in G$,

$q$.insert($g$): $q$ is not full
$q$.maxItem: $q$ is not empty
$q$.minItem: $q$ is not empty
$q$.deleteMax: $q$ is not empty
$q$.deleteMin: $q$ is not empty
$q$.deleteAllMax: $q$ must not be empty.
(Operations without preconditions are omitted)

**Semantics:** For all $n \in N$, $g \in G$, $n \in \mathbb{N}$

newPriorityQueue<G>($n$): create a new queue with capacity $n$.
$q$.insert($g$): insert item $g$ into $t$ in priority order with the highest number being the highest priority.
$q$.isFull: return *true* if $t$ is full, *false* otherwise
$q$.isEmpty: return *true* if $t$ is empty, *false* otherwise
$q$.maxItem: return the largest (highest priority) item in $q$.
$q$.minItem: return the smallest (lowest priority) item in $q$.
$q$.deleteMax: remove the largest (highest priority) item in $q$ from $q$.
$q$.deleteMin: remove the smallest (lowest priority) item in $q$ from $q$.
$q$.frequency(g): return number of times element $g$ appears in the queue regardless of priority.
$q$.deleteAllMax: all occurrences of the highest priority item are deleted from $q$.

### Alternative Answers

For the `insert` operation it is reasonable to not assume that the priority of the item is embedded in the item $g$, since we don't assume any particular implementation. Thus, the following alternative signature, precondition, and semantics for `insert`, respectively, are acceptable:

$Q$.insert($g$,$p$): $G \times \mathbb{N}_0 \not\to Q$
$q$.insert($g$,$p$): $q$ is not full
$q$.insert($g$,$p$): insert item $g$ with priority $p$ into $t$ in priority order with the highest number being the highest priority.

It would also be acceptable to allow $p$ to be from the set of integers, i.e. $\mathbb{Z}$, but this would require that $\mathbb{Z}$ be defined in the "sets" section of the specification.

However, it does not make sense for any other operations to receive a priority. value as input.