| **Queue ADT** |
| --- |
| Priority Queue = { (, , ,...,), min/max} |
| 0 ≤ n ∧ Size(Priority Queue) = n ∧ min =element with the lowest priority in the queue ∧ max = element with the highest priority in the queue |
| * Priority Queue − → Priority Queue * enqueue Priority Queue × Element → Priority Queue * dequeue Priority Queue → Element * peek Priority Queue → Element * isEmpty Priority Queue → Boolean * size Priority Queue → Integer |

| **Priority Queue − → Priority Queue**  Builds an empty priority queue  Preconditions: −  Postconditions: Priority Queue pq = ∅. |
| --- |

| **enqueue Priority Queue × Element → Priority Queue**  Inserts a new element e to the back of the priority queue pq  Preconditions: Priority Queue pq = (, , ,..., ) and element e or pq = ∅ and element e  Postconditions:Priority Queue pq = (, , ,..., , ) or pq = () |
| --- |

| **dequeue Priority Queue → Element**  Remove and return the element with the highest priority in the priority queue.  Preconditions:Priority Queue pq ∅ i.e. s = (, , ,..., )  Postconditions: Priority Queue q = (, , ,..., ) and Element returned is emax (if the element with the highest priority was removed) or emin (if the element with the lowest priority was removed), where emax or emin is the element with the highest or lowest priority, respectively, in the priority queue before the operation." |
| --- |

| **peek Priority Queue → Element**  Return the element with the highest priority in the priority queue without removing it.  Preconditions:Priority Queue pq ∅ i.e. s = (, , ,..., )  Postconditions: Element returned is emax (if the element with the highest priority is returned) or emin (if the element with the lowest priority is returned), where emax or emin is the element with the highest or lowest priority, respectively, in the priority queue." |
| --- |

| **isEmpty Priority Queue → Boolean**  Determines if the priority queue pq is empty or not  Preconditions: Priority Queue pq  Postconditions: True if q = ∅, False if q ∅ |
| --- |

| **size Priority Queue → Integer**  Returns the number of elements currently stored in the priority queue .  Preconditions: Priority Queue pq  Postconditions: size(pq) = n, where n is the number of elements in the priority queue . |
| --- |