TAD <PriorityQueue>
$A = [a_{[0]}, \ldots, a_{[n-1]}]$
- {inv: $0 \leq i <$A.getSize()} - {inv: $a_{[\lfloor(i-1)/2\rfloor]}$ is father of $a_{[i]}$} - {inv: $a_{[i*2+1]}$ is left son of $a_{[i]}$} - {inv: $a_{[i*2+2]}$ is right son of $a_{[i]}$}
Primitive operations: - PriorityQueue constructor PriorityQueue → PriorityQueue - insert modifier PriorityQueue x value → PriorityQueue - maximum analyzer PriorityQueue → value - extractMax modifier PriorityQueue → value - increaseKey modifier PriorityQueue x value x key → PriorityQueue

PriorityQueue()
"Creates and returns a new empty priority queue."
pre: none
pos: creates an empty priority queue.

insert(value)
"Adds the element to the priority queue."
pre: none
pos: adds new element at the end of the queue.

maximum()
"Returns the element with the maximum priority in the priority queue without removing it from the queue."
pre: priority queue is not empty. Otherwise, throws an exception.
pos: returns the item at the front of the priority queue.

extractMax()
"Removes and returns the element with the maximum priority in the priority queue."
pre: priority queue is not empty. Otherwise, throws an exception.
pos: removes and returns the item at the front of the priority queue. Likewise, it reorders the priority queue.

increaseKey(value, key)
"Increases the priority of the element in the priority queue to the new value of key."
pre: the index must be within the range and the new key value has been greater than the current value.
pos: the value of the key in the specified index is increased to the value provided.