Heap Complexities

Running Times			
	Binary Heap	Sorted List/Array	Unsorted List/Array
min()	$\Theta(1)$	$\Theta(1)$	$\Theta(n)$
insert() (worst case)	$\Theta(logn)^*$	$\Theta(n)$	$\Theta(1)^*$
insert() (best case)	$\Theta(1)^*$	$\Theta(1)^*$	$\Theta(1)^*$
removeMin() (worst case)	$\Theta(logn)$	$\Theta(1)$	$\Theta(n)$
removeMin() (best case)	$\Theta(1)$	$\Theta(1)$	$\Theta(n)$

* If you are using an array-based data structure, these running times assume that you don't run out of room. If you do, it will take $\Theta(n)$ time to allocate a larger array and copy the entries into it. However, if you double the array size each time, the average running time will still be as indicated.