

2) Sorted in increasing order
 $O(n \log n)$

Because it is rearranged into a MaxHeap using BuildMaxHeap which takes $O(n)$ time, then sorted back into increasing order using the for loop which runs $n-1$ times, & calls MaxHeapify which runs $\log n$ times. This means that Heapsort takes $O(n \log n)$ time.

Sorted in decreasing order

This would only affect the time it takes to build the MaxHeap & not the time it takes to sort it inside the for loop, so it is also $O(n \log n)$ for the same reasons as increasing order