6.4-3.

What is the running time of HEAPSORT on an array A of length n that is already sorted in increasing order? What about decreasing order?

Answer.

The running time of HEAPSORT on an array A of length n that is already sorted in increasing order is $\Theta(n \lg n)$. HEAPSORT first transforms the array into a max-heap, this takes $O(n)$ time (see Section 6.3). Then the **for** loop successively removes $n-1$ largest element from the heap, where each time MAX-HEAPIFY is called, walking over the full height of the tree. So the process generated by **for** loop is $\Theta(n \lg n)$, making the HEAPSORT procedure have an order of growth $\Theta(n \lg n)$.

The running time of HEAPSORT on an array A of length n that is already sorted in decreasing order is also $\Theta(n \lg n)$, since the process **for** loop generates here is exactly the same to the former case.

*. Creative Commons 2014, Lawrence X. Amlord (颜世敏, aka 颜 序).
Email address: informlarry@gmail.com