1. (CLRS 12.2-5) Show that if a node in a binary search tree has two children, then its successor has no left child and its predecessor has no right child.
2. (CLRS 12-2 - Radix trees) Show how to use a radix tree to sort $S$ lexicographically in $O(n)$ time.
3. (CLRS 9-1) Given a set of \( n \) numbers, we wish to find the \( i \) largest in sorted order using a comparison-based algorithm. Find the algorithm that implements each of the following methods with the best asymptotic worst-case running time, and analyze the running times of the algorithms on terms of \( n \) and \( i \).

(a) Sort the numbers, and list the \( i \) largest.
(b) Build a max-priority queue from the numbers, and call EXTRACT-MAX \( i \) times.
(c) Use an order-statistics algorithm to find the \( i \)th largest number, partition around the number, and sort the \( i \) largest numbers.
4. \((extra\;credit)\) (CLRS 8-2)