CSC 225 SUMMER 2016
ALGORITHMS AND DATA STRUCTURES I
ASSIGNMENT 3
UNIVERSITY OF VICTORIA

1. Consider a version of the deterministic quick-sort algorithm that uses the element at rank $\lfloor n/2 \rfloor$ as the pivot for a sequence on n elements.

   (a) What is the running time of this version on a sequence that is already sorted?
   (b) Describe the kind of sequence that would cause this version of quick-sort to run in $\Theta(n^2)$ time.

2. Illustrate the performance of the heap-sort algorithm on the following input sequence, *S*, by drawing the heap tree after each insert() and removeMin() call. That is, there should be 20 trees, each the final result of the given operation after bubbling is complete.

$$S = (2, 5, 16, 4, 10, 23, 39, 18, 26, 15)$$

3. Design algorithms for the following operations for a node *v* in a binary tree *T*:

   (a) preorderNext(v): return the node visited after *v* in a preorder traversal of *T*.
   (b) inorderNext(v): return the node visited after *v* in an inorder traversal of *T*.
   (c) postorderNext(v): return the node visited after *v* in a postorder traversal of *T*.

   What are the worst-case running times of your algorithms?

4. Define the ***internal path length***, $I(T)$, of a tree *T* to be the sum of the depths of all the internal nodes in *T*. Likewise, define the ***external path length***, $E(T)$, of a tree *T* to be the sum of the depths of all the external nodes in *T*. Show that if *T* is a proper binary tree with *n* internal nodes, then $E(T) = I(T) + 2n$.

5. Suppose we are given a sequence *S* of *n* elements, each of which is an integer in the range $[0, n^3 - 1]$. Describe a simple method for sorting *S* in O(*n*) time.

   Hint: Think of an alternate way of representing the elements.