1. **Exercise 7.1-1 (p. 148) (30 Points)**
   Using Figure 7.1 as a model, illustrate the operation of \textsc{Partition} on the array
   \(A = [13, 19, 9, 5, 12, 8, 7, 4, 11, 2, 6, 21]\).

2. **Exercise 7.2-2 (p. 153) (10 Points)**
   What is the running time of \textsc{Quicksort} when all elements of array \(A\) have the same value.

3. **Problem 7-3 Stooge sort (p. 161) (30 Points)**
   Professors Howard, Fine, and Howard have proposed the following “elegant” algorithm:

   \[
   \text{STOOGE-SORT}(A, i, j) \\
   \text{1. if } A[i] > A[j] \\
   \text{2. then exchange } A[i] \leftrightarrow A[j] \\
   \text{3. if } i + 1 \geq j \\
   \text{4. then return} \\
   \text{5. } k \leftarrow \lfloor (j - i + 1)/3 \rfloor \quad \triangleright \text{Round down.} \\
   \text{6. STOOGE-SORT}(A, i, j - k) \quad \triangleright \text{First two-thirds.} \\
   \text{7. STOOGE-SORT}(A, i + k, j) \quad \triangleright \text{Last two-thirds.} \\
   \text{8. STOOGE-SORT}(A, i, j - k) \quad \triangleright \text{First two-thirds again.}
   \]

   (a) **You are not responsible from Part (a)** Argue that \textsc{Stooge-Sort}(\(A, 1, \text{length}[A]\))
correctly sorts the input array \(A[1..n]\), where \(n = \text{length}[A]\).

   (b) Give a recurrence for the worse-case running time of \textsc{Stooge-Sort} and a tight asymptotyc \((\Theta\text{-notation})\) bound on the worse-case running time.

   (c) Compare the worse-case running time of \textsc{Stooge-Sort} with that of insertion sort, merge
sort, heapsort, and quicksort. Do the Professors deserve tenure?

4. **Exercise 8.2-4 (p. 170) (30 Points)**
   Describe an algorithm that, given \(n\) integers in the range 0 to \(k\), preprocesses its input and
then answers any query about how may of the \(n\) integers fall into a range \([a..b]\) in \(O(1)\) time.
Your algorithm should use \(\Theta(n + k)\) preprocessing time.