Homework 4: Sorting

You will submit your solution to this assignment to the Curator System (as HW4). Your solution must be either a plain text file (e.g., NotePad) or a typed MS Word document; submissions in other formats will not be graded. Credit will only be given if you show relevant work.

1. [15 points] Here is an array which has just been partitioned by the first step of Quicksort:

   3, 0, 2, 4, 5, 8, 7, 6, 9

   Which of these elements could have been the pivot? (if there are more than one possibility, list them all)

2. [25 points] Put the following list of values into the order that would produce the worst possible case when input into Quicksort. Assume Quicksort is using the partitioning algorithm as shown in the course notes, including the pivot selection algorithm that selects the middle-most element (rounded down in the case of lists of even size) as in \( \text{Pivot} = \text{List}[\lfloor(\text{Lo} + \text{Hi})/2\rfloor] \). Then make a sequence showing the state of the array after each recursive level of the Quicksort algorithm, marking the chosen pivots and the recursive subdivision of the array.

   1, 2, 3, 4, 5, 6, 7, 8, 9

3. [30 points] Prove by strong induction that sorting \( n \) elements with MergeSort takes \( T(n) = n \cdot \log_2 n \) steps, for all \( n \) that are powers of 2. Assume the \textit{merge} algorithm within the MergeSort algorithm takes exactly \( m \) steps to merge a total of \( m \) elements from two sorted sub-lists, and that all other atomic operations take 0 steps.

   Hint: you will need to define a recursive equation for \( T(n) \) based on the logic of MergeSort.

4. [30 points] Write an algorithm that takes an unsorted list of \( n \) integers whose values are in the range of 0 to \( n \), and outputs a list of all duplicate integers. There must be no duplicates in the output list, which can also be unsorted. The algorithm must run in \( \Theta(n) \) time. \textit{Show your analysis of the running time.}