CSCI 4041, Fall 2018, Written Assignment 1

Due Thursday, 9/13/18, 10:30 AM (submission link on Canvas)

This is a collaborative assignment; you may work in a group of 1-3 students. However, you may not consult or discuss the solutions with anyone other than the course instructor, the TAs, or the other members of your group, nor may you use material found from outside sources as part of your solutions. In addition, if you do choose to work in a group, each group member must participate in coming up with the solution to each problem, and must be able to explain the group's answer if asked: dividing the problems amongst the group members is not acceptable.

Complete the following problems and submit your solutions in a single pdf file to the Written Assignment 1 submission link on Canvas. If you're working in a group, only one person should submit your answers, but make sure that you include the name and x500 of each group member at the top of the file, and that you are all in one of the WA1 Groups in Canvas. Typed solutions are preferred, but pictures or scans of a handwritten solutions in pdf form are acceptable so long as your solutions are clearly legible.

This assignment contains 3 problems, and each is worth 10 points, for a total of 30 points.

Your solutions to these problems must be clearly explained in a step-by-step manner; for most problems, the explanation will be worth far more points than the actual answer.

1. (Problem 3-4f in the textbook) Let f(n) and g(n) be asymptotically positive functions. Prove or disprove the following: $f(n) = O(g(n))$ implies $g(n) = \Omega(f(n))$. You **must** use the definition of big-O and big-Ω notation, given on pages 47 and 48 of the textbook, to construct your argument.

2. (Adapted from Exercise 2.2-2 in the textbook) Below is pseudocode for selection sort, which works by finding the minimum element of A, and storing it in A[1], then finding the second smallest element of A, and storing it in A[2], and so on.

```
SELECTION-SORT(A)
1     for i = 1 to A.length-1
2           for j = i+1 to A.length
3                 if A[j] < A[i]
4                       exchange A[i] with A[j]
```

   a. State precisely a loop invariant for the inner for loop in lines 2-4, and prove that this loop invariant holds. Your proof should resemble the structure of the loop invariant proofs in Sections 2.1 and 2.3 of the textbook, including arguments for Initialization, Maintenance, and Termination.

b. Using your answer from part a, state precisely a loop invariant for the outer for loop in lines 1-4, and prove that this loop invariant holds.

c. To prove that Selection-Sort is correct, we must prove that the final list A' is a permutation (reordering) of the original list A such that A'[1] ≤ A'[2] ≤ … ≤ A'[n]. Your answer from part b should prove that A'[1] ≤ A'[2] ≤ … ≤ A'[n], so explain why we know that the final list A' is a permutation (reordering) of the original list A.

d. Give a best-case and worst-case running time for selection sort using Θ-notation, and argue informally that your answer is correct.

3. Using Figure 2.4 from the textbook as a model (see below), illustrate the operation of merge sort on the array A = [7, 18, 1, 22, 9, 20, 25, 6].

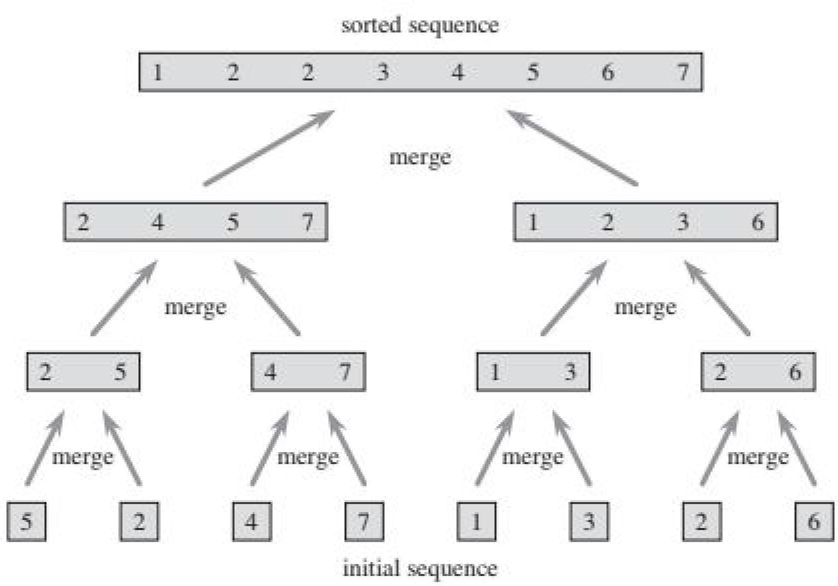

Figure 2.4 The operation of merge sort on the array $A = \langle 5, 2, 4, 7, 1, 3, 2, 6 \rangle$. The lengths of the sorted sequences being merged increase as the algorithm progresses from bottom to top.