CSCI 3322 (Principles of Algorithms), Fall 2022
Homework 1

Credit: 52.5 points.

1 Reading

Be sure you have read, or at least skimmed, Chapter 2 of the textbook.

2 Problems

Answer the following questions. You may write out your answers by hand and scan them, or you may use a word processor or other program, but please turn in a PDF or plain text file. (No links to shared files on Google Drive please, and no word-processor files.) Turn it in by putting it in your course "TurnIn" folder on Google Drive. Please be sure to include your name somewhere in the file, so when I print it for grading I know whose work it is. (In the pledge is fine.)

1. Consider the *searching problem*, which you probably saw in CS1:

   Input: A sequence of n numbers $a_1, a_2, \ldots a_n$ stored in an array $A[1:n]$ and a value x.

   Output: An index i such that $x = A[i]$, or the special value NIL if x does not appear in A.

   One way to solve this problem is with *linear search* algorithm, which scans through the array from beginning to end.

   - (5 points) Write pseudocode for an iterative algorithm for linear search.
   - (5 points) Use a loop invariant to argue that your algorithm is correct, as the textbook does for INSERTION-SORT.
   - (5 points) Analyze the worst-case execution time for your algorithm.
   - (5 points) Write pseudocode for a recursive version of the same algorithm.

2. Another way to solve the same problem, possible if A is sorted, is *binary search*: Check the midpoint of A against the value to find (x). You can then eliminate half the array from further consideration — if x is less than the midpoint of the array, you need only search the left part of the array, while if x is larger than the midpoint, you need only search the right part.

   - (5 points) Write pseudocode for an algorithm, either iterative or recursive, for binary search. (I'm inclined to think recursion is easier, but either one is okay here.)
     item (5 points) Argue that worst-case running time of your algorithm is $\Theta(\log n)$.

3. (10 points) You probably remember the bubblesort algorithm from CS1. The textbook presents pseudocode for this algorithm on p. 46.

   - (2.5 points) In order to show that BUBBLESORT(A, n) is correct, you need to show that it terminates and that
     $A'[1] \leq A'[2] \leq \cdots \leq A'[n]$.
     Is that enough, or is there something else you also need to show? What?

- (5 points) State a loop invariant for the outer loop and argue that meets the conditions for being an invariant of the loop.
- (5 points) Argue that worst-case running time of your algorithm is $\Theta(n^2)$.

2.1 Pledge

This should include the Honor Code pledge, or just the word "pledged", *plus* at least one of the following about collaboration and help (as many as apply). Text *in italics* is explanatory or something for you to fill in; you don't need to repeat it!

- I did not get outside help *aside from course materials, including starter code, readings, sample programs, the instructor.*
- I worked with *names of other students* on this assignment.
- I got help with this assignment from *source of help — ACM tutoring, another student in the course, etc. (Here, "help" means significant help, beyond a little assistance with tools or compiler errors.)*
- I got help from *outside source — a book other than the textbook (give title and author), a Web site (give its URL), etc.. (Here too, you only need to mention significant help — you don't need to tell me that you looked up an error message on the Web, but if you found an algorithm or a code sketch, tell me about that.)*
- I provided help to *names of students* on this assignment. *(And here too, you only need to tell me about significant help.)*

2.2 Essay

This should be a brief essay (a sentence or two is fine, though you can write as much as you like) telling me what if anything you think you learned from the assignment, and what if anything you found interesting, difficult, or otherwise noteworthy.