

Analysis of Algorithms
CS 375, Spring 2020

Homework 7

Due **AT THE BEGINNING OF CLASS** Wednesday, March 4

- From your textbook (CLRS), please read all of Chapters 2 and 3. Although not all of it will show up on CS375 exercises, it is all good material to know.
- Please recall the guidelines for algorithms given on previous HW assignment sheets. They continue to apply to all HWs in CS375.
- In general, there may be multiple correct ways of presenting an algorithm, although excessively inefficient or inelegant solutions may not receive full credit. If you have questions about whether your proposed solution is excessively inefficient or inelegant, please ask your Prof.!
- *A general note for CS375:* When writing up your homework, please write neatly and **explain your answers clearly**, giving all details needed to make your answers easy to understand. Graders may not award full credit to incomplete or illegible solutions. Clear communication *is* the point, on every assignment.

In general in CS375, unless explicitly specified otherwise, answers should be accompanied by explanations. Answers without explanations may not receive full credit. Please feel free to ask me any questions about explanations that might come up!

Exercises

1. **Recursive Insertion Sort!** In this exercise, you'll write a pseudocode algorithm for a recursive version of Insertion sort, a different way of expressing the same underlying algorithmic idea as the iterative version from class.

We'll do this in two parts, ending up with an algorithm to sort LLists of numbers (i.e., using the LList data structure from class for the sequence to be sorted). For this exercise, sorting is taken to mean in non-decreasing order.

- (a) Write a recursive *LLInsert* algorithm that inserts a number x in the proper location in a sorted LList L .

```
# Input: Number  $x$  and sorted LList  $L = [a_0, a_1, \dots, a_n]$ ,  
#         where  $a_0 \leq a_1 \leq \dots a_n$   
# Output: List  $L' = [b_0, b_1, \dots, b_{n+1}]$  containing input  $x$  and the  
#          $n + 1$  elements of  $L$ , in sorted order  
#          $b_0 \leq b_1 \leq \dots \leq b_{n+1}$ 
```

- (b) Using the *LLInsert* function, write a recursive *LLInsertionSort* algorithm that takes an LList L of numbers, possibly unsorted, and returns a sorted LList L' with the same elements as L in sorted order, consistent with the specification of the sorting problem.

```
# Input: LList  $L = [a_0, a_1, \dots, a_n]$ 
# Output: List  $L' = [b_0, b_1, \dots, b_n]$  containing exactly the
#         elements of  $L$ , in sorted order  $b_0 \leq b_1 \leq \dots \leq b_n$ 
```

As usual, explain your algorithms and give correctness arguments.

HW8 Lookahead!

The following exercise is going to be part of HW8, which will be assigned on Wednesday. It is included here for you to look at, to get a head start on it if you are comfortable with the needed mathematical background, or to ask questions about it if you have any questions about it! **Do not turn in this exercise with HW7—please submit it as part of your HW8, after it's assigned there!**

- List the following functions of n according to their order of growth—that is, how fast each function grows as n gets big—from lowest to highest:

$$(n-2)!, \quad 5 \lg(n+100)^{10}, \quad 2^{2n}, \quad 0.001n^4 + 3n^3 + 1, \quad \ln^2 n, \quad \sqrt[3]{n}, \quad 3^n.$$

(As is conventional, the $\lg$ function is logarithm base 2; the $\ln$ function is the *natural logarithm*, logarithm base e ; and $\ln^2 n$ is common notation for $(\ln n)^2$.) Although you don't need to explain every part of the ordering for this exercise, please give short explanations (1–2 sentences) for the following:

1. how you know the second-smallest comes before the third-smallest; and
2. how you know the second-largest comes after the third-largest.