CS 430 Introduction to Algorithms

Spring Semester, 2021

Homework 1 version 1.1

Assigned: February 1

Due: February 15

Please respect the following guidelines for writing pseudocode:

- 1. C/Java/Python instructions are fine. But do not write object-oriented additions. Do not declare or use any class. Declare only procedures (if necessary) and explain in words what each procedure does, and what is the use of each parameter.
- 2. One instruction per line
- 3. Match the brackets with a horizontal line, or use indentation
- 4. Number your lines
- 5. Write down if your array is indexed $0 \dots n-1$ or $1 \dots n$. Problem 4 forces you to use $1 \dots n$.

Problem 1 Order the following list of functions by the big-Oh notation. Group together (for example, by underlining) those functions that are big-Theta of one another.

4^n	n^3	$n^2 \log n$	4^{logn}	\sqrt{logn}
2^{2^n}	$\left\lceil \sqrt{n} \right\rceil$	$n^{0.01}$	1/n	$4n^{3/2}$
$3n^{0.5}$	5n	$\lfloor 2n log^2n \rfloor$	2^n	$n\log_4 n$
$6n\log n$	2^{100}	$\log \log n$	$\log^2 n$	2^{logn}

Hint: When in doubt about two functions f(n) and g(n), consider $\log f(n)$ and $\log g(n)$ or $2^{f(n)}$ and $2^{g(n)}$.

Problem 2 Matrix multiplication. Write pseudocode for computing $C = A \times B$, where A, B, C are matrices of integers of size $n \times m$, $m \times k$, and $n \times k$ respectively.

Analyze the running time of your algorithm, as a function of n, m, and k.

Problem 3 You are given a tree represented by the following data structure. Nodes have three fields: element, left-child, and right-child. The element is a integer.

Write a procedure (pseudocode) with input parameter the location of the root node (or pointer to the root node), and that computes the average value of the elements in the tree. However, do not use recursion. Use a stack instead.

Analyze the running time of your algorithm, as a function of the number of nodes in the tree. As proven in CS 331, stack operations are O(1).

Problem 4 Give pseudocode for the following problem: given an array A[1...n] containing the numbers 1, 2, ..., n and representing a permutation, modify A to represent the next lexicographic permutation.

Analyse running time (which should not exceed O(n)) and argue correctness (why the output is lexicographically bigger than the input? why other permutations lexicographically bigger than the input are bigger than the output?).

Problem 5 What is the running time of HEAPSORT on an array A that is sorted in decreasing order? Use an $\Omega()$ approximation and justify you answer.