Home Exercise 1: Combinatorics, O-notation, and Data Structures

Algorithms and Complexity lecture
at CentraleSupélec / ESSEC

Dimo Brockhoff
firstname.lastname@inria.fr

due: September 25, 2020

Abstract

Please send your solutions by email to Dimo Brockhoff (preferably in PDF format) with a clear indication of your full name until the submission deadline on September 25, 2020 (a Friday). Groups of 5 students are explicitly allowed and highly encouraged. In the case of group submissions, please make sure that you submit maximally three times with the same partner!

1 Matrix Multiplication (5 points)

How many additions and how many multiplications does the well-known algorithm from linear algebra needs to multiply a matrix of size $m \times n$ with another one of size $n \times l$?

2 Finding Smallest Elements (5 points)

Assume you are given a (potentially unsorted!) array A of n elements. Describe an algorithm that computes the smallest element in A and how long it takes to compute it in the best *and* the worst case.

3 Finding Smallest Elements II (5 points)

How must your algorithm change (and how does its runtime change) when, instead of the smallest element, you are supposed to find the smallest 2 or the smallest 3 elements in an array A of n elements? Assume that $n >> 3$.

4 O notation (5 points)

Which of the following "equations" hold? Please give either a proof or a counter example. Note that $O(f_1) \circ O(f_2)$ denotes the set $\{g_1 \circ g_2 \,|\, g_1 \in O(f_1), g_2 \in O(f_2)\}$ for any operator $\circ$ between two functions g_1 and g_2.

1. $O(f_1) = O(\log(f_1))$
2. $O(f_1) \cdot O(f_2) = O(f_1 \cdot f_2)$