University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
S. Faour, M. Fuchs, G. Schmid

Algorithm Theory
Exercise Sheet 1

Due: Wednesday, 26th of October, 2022, 11:59 pm

Exercise 1: Landau Notations *(5 Points)*

Prove or disprove the following:

(a) $n \log n \in \Theta(\log(n!))$ *(2 Points)*

(b) $n^3 - 9n^2 \in \Omega(n^3)$ *(1 Point)*

(c) $\big(\log(\sqrt{n})\big)^2 \in \Theta(\log n)$ *(2 Points)*

Exercise 2: The Majority Element *(7 Points)*

An array $A[1...n]$ is said to have a *majority element* if more than half of its elements are the same. Given an array, the goal is to design an efficient algorithm to tell whether the array has a majority element or not, and, if so to find that element.
Note that the elements of the array need not to belong to some ordered domain e.g. $\mathbb{N}$, hence there can be no comparisons of the form e.g. $A[i] < A[j]$. However, we can test equality of two elements in $O(1)$-time.

(a) Give an algorithm that solves the problem in $O(n \log n)$-time, argue correctness, and analyze its running time. *(2 Points)*

(b) Try speeding things up and give a linear time algorithm that solves the problem, argue correctness, and analyze its running time (don't forget to write down the recurrence relation). *(5 Points)*
Hint: try reducing the size of the array to at most half via pairing up elements.

Exercise 3: Almost Closest Pairs of Points *(8 Points)*

In the lecture, we discussed an $O(n \log n)$-time divide-and-conquer algorithm to determine the closest pair of points. Assume that we are not only interested in the closest pair of points, but in all pairs of points that are at distance at most twice the distance between the closest two points.

(a) How many such pairs of points can there be? It is sufficient to give your answer using big-O notation. *(3 Points)*

(b) Devise an algorithm that outputs a list with all pairs of points at distance at most twice the distance between the closest two points. Describe what you have to change compared to the closest pair algorithm of the lecture and analyze the running time of your algorithm. *(5 Points).*