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

# Algorithms and Datastructures
# Exercise Sheet 6

## Exercise 1: Minimum Distance between Values *(10 Points)*

(a) Given an array $A$ that contains $n$ integers. Describe an algorithm that finds indices $i \neq j$ such that $|A[i] - A[j]|$ is minimal among all indices. In other words, the algorithm should compute the entries of $A$ that have the smallest distance. Argue the correctness of your algorithm and show that it runs in time $o(n^2)$. *(5 Points)*

(b) Now, assume that the $n$ numbers from a) are given in a binary search tree $B$ (instead of in an array). Again, give an algorithm that finds the two tree nodes $u \neq v$ such that $|\text{val}(v) - \text{val}(u)|$ is minimal. Show the correctness and explain why the runtime is on $O(n)$. *(5 Points)*

## Exercise 2: *(10 Points)*

Again, given a binary tree $B$ containing $n$ integers. For a path $P = \{r, v_1, v_2, \ldots, b\}$, from the root node $r$ to some leaf $b$, we define its weight by $w(P) = \sum_{v \in P} \text{val}(v)$. Describe an algorithm that finds the *heaviest* path from the root node to some leaf in $B$, i.e., the path $P$ that maximizes $w(P)$ for all root-to-leaf path. State that the runtime is in $O(n)$. *(10 Points).*