Marburg University

Department of Computer Science
Prof. Dr. Sebastian Wild

Date: 2026-05-01
Version: 2026-04-30 17:57

Exercise Sheet 2 for Advanced Data Structures (Summer 2026)

Hand In: *Until 2026-05-08 18:00, on ILIAS.*

Problem 1

30 points

Recall the "move-to-root" heuristic in binary search trees: when accessing an element, this heuristic repeatedly rotates the edge between that element and its parent until the accessed element becomes the root.

Give an example where the amortized access time for this heuristic in a tree with n vertices is $\omega(\log n)$. For full marks, amortized costs should be $\Omega(n)$.

Problem 2

30 points

Splay trees have the *static optimality property*. Suppose that we construct a splay tree on elements $x_1 < \cdots < x_n$, and that each element x_i is accessed $a_i > 0$ times. Suppose $\sum_i a_i = N$. Then, static optimality says that the cost to access all of these elements is at most

$$O(N + \sum_i a_i \log(N/a_i)). \tag{1}$$

Prove that treaps do not satisfy the static optimality property: there exists some access sequence to a treap which uses asymptotically strictly more than (1) operations in expectation.