Exercise sheet 6.

Data structures SoSe 2020

László Kozma, Katharina Klost

Due 12:00, June 5th, 2020

Exercise 1 Doubling search *4 Points*

Recall the doubling binary search method from class, with running time $2\log_2 d$, where d is the *rank* of the searched element. Show that this running time can be improved to $\log_2 d + o(\log d)$.

Bonus question (+3p): What is the best you can achieve?

Exercise 2 Adaptive sorting *3+3 Points*

Given is a list $A = (a_1, \ldots, a_n)$ of elements from an ordered set. We want to show that A can be sorted faster than the worst case $\Theta(n \log n)$, if it has some structure. Show that A can be sorted in time:

(a) $O(n \log k)$, if A consists of k increasing *runs*. A run is a contiguous increasing subsequence $a_i \leq a_{i+1} \leq \cdots \leq a_j$.

(b) $O(n \log d)$, where d is the *average rank difference* between neighboring elements. The rank difference between two neighbors a_i and a_{i+1} is the number of elements in A that are at least $\min(a_i, a_{i+1})$ and at most $\max(a_i, a_{i+1})$.

Exercise 3 Jordan sequence *2 Points*

A Jordan sequence is obtained by numbering the intersections between a curve (no self-intersections) and a horizontal line, in the order they appear on the curve, and reading the numbers out in left-to-right order on the line. Show that the following sequence is a Jordan sequence. Give an example short sequence that is not a Jordan sequence.

$11, 1, 10, 7, 4, 3, 8, 9, 2, 12, 13, 5, 6, 14, 16, 15$

Exercise 4 Programming exercise *(30 Points altogether)*

This exercise is ongoing for 3 weeks (deadline: June 29th). Please decide on a topic.

Total: 12 points. Have fun with the solutions!