Exercise sheet 7.

Data structures

SoSe 2020

László Kozma, Katharina Klost

Due 12:00, June 12th, 2020

Exercise 1 Implementation of finger search trees *5 Points*

Describe precisely the bookkeeping necessary to implement a finger search tree based on a $(2, 4)$-tree (all pointers/fields you need to store in each node and any other auxiliary variables you may need).

Give a pseudocode implementation of *finger search*, and one (or both) of *insert* and *delete*, describing all necessary updates and pointer moves.

Bonus question (+3p): Give a pseudocode implementation of splitting a finger search tree at x.

Exercise 2 Amortized analysis of (a, b)-trees *4 + 2 Points*

(a) Prove that a list of k insert and ℓ delete operations in an initially empty $(2, 4)$-tree takes total time $O(k + \ell)$, assuming that the pointers to the insert/delete location are always given, so no search is necessary. (Insert and delete operations can be arbitrarily intermixed.)
Hint: Observe how the node degrees change and define a suitable potential function.

(b) Show that in a $(2, 3)$-tree this is not the case, and every operation may take $\Omega(\log n)$ time, where n is the current size of the tree. (Give a concrete sequence of operations that shows this.)

Exercise 3 Programming exercise *1 Point*

The programming exercise is due June 29th (30 points).

For this time, please write a brief summary (one paragraph) of your plan and/or progress so far.

Total: 12 points. Have fun with the solutions!