**Lecture** We will talk about amortised analysis of splay trees. You should read

- Splay trees in section 16.5-16.6 in the notes by Jeff Ericsson.

Supplementary material

- Self-adjusting binary top-trees by Sleator and Tarjan.

## Exercises

### 1 Splay trees

**1.1** Show the splay tree that results from inserting the keys 41, 38, 31, 12, 19 , 8 in this order into an initially empty tree.

**1.2** Show how the tree looks after deleting 31.

**2 Play trees** Professor Bille suggests a simpler version of Splay Trees that he calls Play trees. Play Trees are similar to Splay Trees but uses only single rotations when splaying.

**2.1** What is the amortized cost of splay(x) if we only use single rotations? Analyze it with the same potential function as we used for Splay Trees.

**2.2** What is the total (actual) cost of first $n$ inserting elements with keys $1, 2, 3, \ldots, n$ in that order in a Play Tree and then searching for $1, 2, 3, \ldots, n$ (in that order)?

**2.3** Professor Bille claims that the amortized cost of the operations insert, search and delete in Play trees is $O(\log n)$. "You just need to use a more sophisticated potential function" he says. Could he be correct?

**3 Splay trees** Let $\Phi$ be the potential function used to analyze splay trees. I.e. $\Phi = \sum_{v \in T} \text{rank}(v)$. Prove that the potential of a complete binary tree is $O(n)$ and that the potential of a rooted path is $O(n \log n)$.

For more exercises on amortised algorithms, see exercises from course 02210 year 2021.