Prof. Dr. Christian Scheideler

Paderborn, November 10, 2017
Submission: November 17, 2017

Fundamental Algorithms

WS 2017

Exercise Sheet 5

Exercise 1:
Let $w(v)$ be the weight of a node v of a Splay tree. Furthermore, let W be the sum of all node weights.

a) Show that the amortized runtime of the delete(i,T)-operation can be bounded by

$$3 \log \frac{W}{w(i)} + 3 \log \frac{W - w(i)}{w(i^-)} + O(1),$$

where i^- denotes the predecessor of i in T. It can be assumed that i is not the minimal element in T.

b) With the help of a), show that for any Splay tree with n elements, $n/2$ deletions can be performed in time $O(n \log n)$.

Exercise 2:

a) Insert the keys of the sequence $(6, 17, 23, 46, 3, 9, 19)$ in order into an initially empty $(2, 4)$-Tree.

b) On the result of a), delete the keys of the sequence $(17, 6, 23, 9)$ in order.

Draw the resulting tree after each insert and delete operation.

Exercise 3:

a) Show Theorem 3.11 on Slide 109.

b) Show Theorem 3.12 on Slide 110 for $(2, 4)$-Trees.

We define the cost of an operation to be the number of structural changes, i.e., the number of splits for an insert operation or the number of merges for a delete operation. For example, an insert operation followed by k splits has cost k.

For each node v of a $(2, 4)$-Tree T, we define tokens(v) depending on the degree of v according to the following table:

degree(v)	1	2	3	4	5
tokens(v)	2	1	0	2	4

By using the potential function $\phi(s) = \sum_{v \in T} \text{tokens}(v)$, show that the amortized cost of an insert and a delete operation is $O(1)$.