Effiziente Algorithmen und Datenstrukturen I

Aufgabe 1

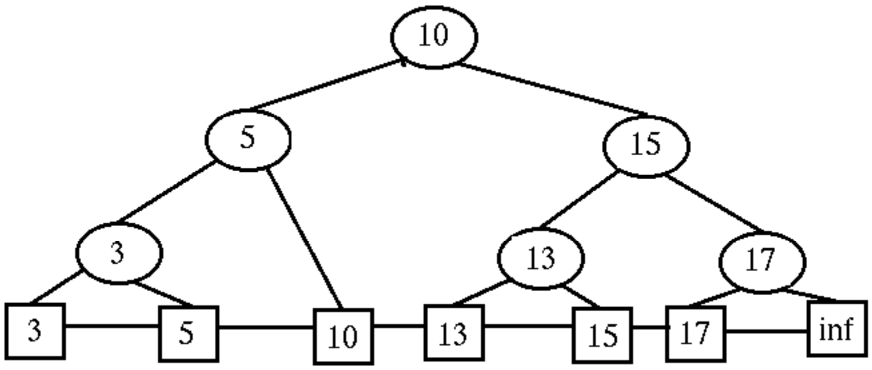
The following is a table of amortized run-time bounds for Radix Heap Operations:

Operation	**Run-Time**
insert	$O(\log C)$
min	$O(1)$
deleteMin	$O(1)$
delete	$O(1)$
decreaseKey	$O(1)$

Prove these amortized run-time bounds. *(**Hint:** Let one time step be the movement of a single element (i.e. $t = \#$ of elements moved). In this case the potential is then represented as the sum of the bin postions + 1 for all elements (i.e. $\Phi = \Sigma_{i=-1}^{K} |B[i]| \cdot (i+1)$))*

Aufgabe 2

Consider the following Splay Tree:

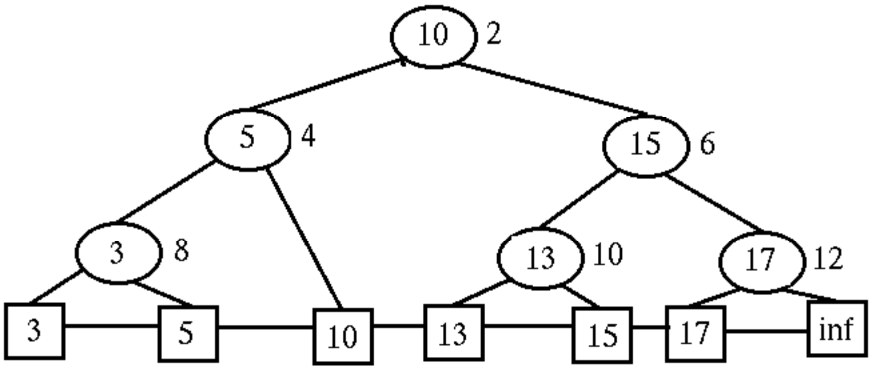


Carry out the operations in the following order and show, after each operation, what the Splay Tree looks like(always carry out each operation on the result of the previous operation):

1. insert(7)
2. delete(5)
3. search(13)

Aufgabe 3

Consider the following Treap (the weights are the values to the right of each element):



Carry out the operations in the following order and show, after each operation, what the Treap looks like(always carry out each operation on the result of the previous operation):

1. insert(8, weight=1)
2. delete(10)

Aufgabe 4

Lemma 3.7: *Let $K = \{k_1, ..., k_n\}$ be a sorted set for which each element k_i has a unique weight $w(k_i)$. In the Treap T corresponding to K, k_j is an ancestor of k_i iff:*

$$w(k_j) = \min\{w(k)|k \in K_{ij}\}$$

where $K_{ij} = \{k_i, ..., k_j\}$.

Prove this lemma.