Technische Universität München
Fakultät für Informatik
Lehrstuhl für Theoretische Informatik
Prof. Dr. Harald Räcke
Richard Stotz, Dennis Kraft

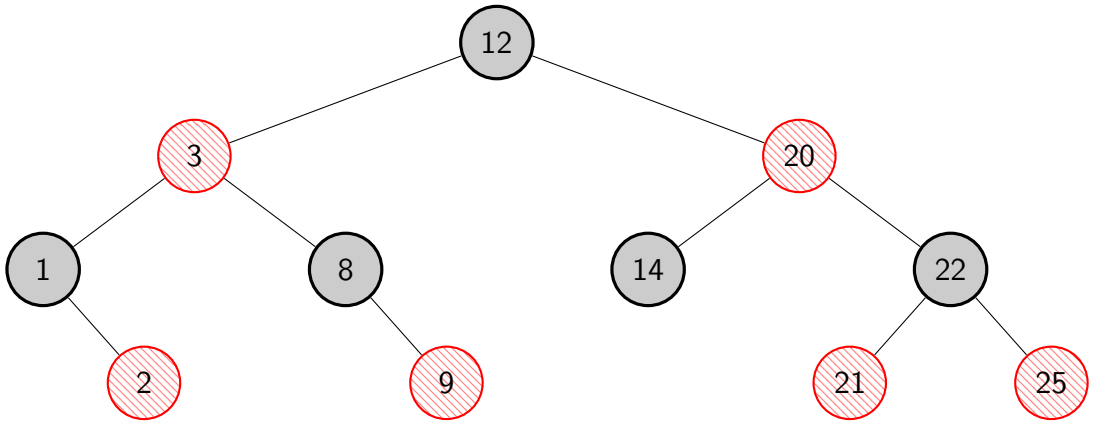
Winter term 2016 / 2017
Assignment 5
November 21, 2016

Efficient Algorithms and Data Structures I

*Deadline: November 28, 10:15 am in the **Efficient Algorithms** mailbox.*

Homework 1 (4 Points)

The teddy bear Alan is asked to decorate the Christmas tree for a major shopping center in the city center of Munich. He chooses to use the glass baubles to form a nice red black tree. Initially, he obtains the following tree:

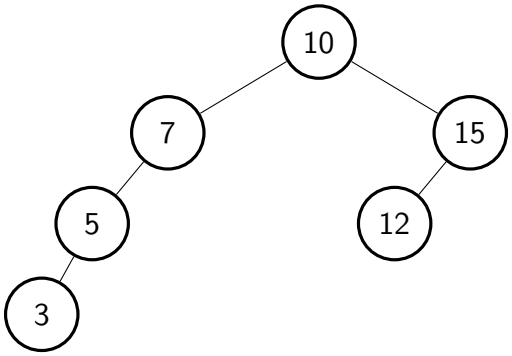


However, during the month of December, several changes must be made to the tree, as some baubles break and others are added to the tree (note that Alan is able to repaint the baubles, if needed).
Show how Alan's tree looks after each change. The tree must remain a red-black tree. Always carry out each operation on the result of the previous operation.

1. Insert 6
2. Insert 10
3. Delete 2
4. Delete 12

Homework 2 (5 Points)

Carry out the following operations sequentially on the splay tree shown below and show what the tree looks like after each operation. Use the splay tree operations defined in the lecture. Always carry out each operation on the result of the previous operation.



1. Search 12
2. Search 3
3. Insert 9

Homework 3 (6 Points)

1. A *complete binary tree* is a binary tree in which every internal node has exactly two children and all paths from the root to a leaf have the same length.

   Let Φ be the potential function used to analyze splay trees, i.e. $\Phi(T) = \sum_{v \in T} r(v)$. Show that the potential of a complete binary tree of $n = 2^k - 1$ nodes is $\mathcal{O}(n)$.

   Hint: You may use without proof that $\sum_{i=0}^{n} i2^i = n2^{n+2} - (n+1)2^{n+1} + 2$.

2. Show that splaying roughly halves the depth of every node on the search path. More precisely, let d be the depth of some node on the search path before the splaying operation and d' its depth after the splaying operation. Show that $d' \leq \lceil d/2 \rceil + 3$ holds.

Homework 4 (5 Points)

A *right-linear chain* is a tree in which every internal node including the root has no left child. Prove or disprove the following statement:
Any binary tree of n nodes can be transformed into a right-linear chain using at most n rotations.

Tutorial Exercise 1

1. Describe how to implement a queue using two stacks and $O(1)$ additional memory, so that the amortized time for any ENQUEUE or DEQUEUE operation is $O(1)$. The only access you have to the stacks is through the standard subroutines PUSH and POP.

2. A quack is a data structure combining properties of both stacks and queues. It can be viewed as a list of elements written left to right such that 3 operations are possible:

   (i) QPUSH: add a new item to the left end of the list

   (ii) QPOP: remove the item on the left end of the list

   (iii) QPULL: remove the item on the right end of the list

   Implement a quack using 3 stacks and $O(1)$ additional memory, so that the amortized time for any QPUSH, QPOP, or QPULL operation is $O(1)$. Again, you are only allowed to access the stacks through the standard functions PUSH and POP.

> Proudest [work]? It's hard to choose. I like the self-adjusting search tree data structure that Danny Sleator and I developed. That's a nice one.
>
> - R. E. Tarjan