Lab 5 – Exercises

EXERCISE 1. Assume that you are given an arbitrary array A of length n and you want to establish the heap property on it by permuting its entries. Consider two procedures for achieving this:

```
buildHeapBackwards =
    for $i = \lfloor n/2 \rfloor$ downto 1 do
        siftDown($i$)

buildHeapRecursive($i \in \mathbb{N}$) =
    if $4i \leq n$
    then
        buildHeapRecursive($2i$)
        buildHeapRecursive($2i + 1$)
    endif
    siftDown($i$)
```

(i) Show that both `buildHeapBackwards` and `buildHeapRecursive`(1) establish the heap property everywhere.

(ii) Implement both algorithms efficiently and compare their running times for random integers and $n \in \{10^i \mid 2 \leq i \leq 8\}$. For this the implementation of `buildHeapRecursive` will be decisive, in particular, you should unravel the recursion for small subtrees.

(iii) Discuss the advantages and disadavantages of the two procedures for the case that data is stored in external storage.

EXERCISE 2. *Pairing heaps* are an efficient data structure using a very simple technique for rebalancing, though a full theoretical analysis is missing. They rebalance only in connection with the *deleteMin* operation. If $r_1, \dots, r_k$ is the sequence of root nodes stored in a location `roots`, then *deleteMin* combines r_1 with r_2, r_3 with r_4, etc., i.e. the roots are paired.

(i) Explain how to implement pairing heaps using three pointers per heap item i: one to the oldest child (i.e. the child linked first to i), one to the next younger sibling (if any), and one to the next older sibling. If there is no older sibling, the third pointer goes to the parent.

(ii) Explain how to implement pairing heaps using only two pointers per heap item: one to the oldest child and one to next younger sibling. If there is no younger sibling, the second pointer goes to the parent.

Figure 6.8 in the [Mehlhorn 2008] textbook (available on BB) illustrates pairing heaps with three or two pointers per heap item.

EXERCISE 3. A *McGee heap* has the same structure as a Fibonacci heap, but supports just the mergeable heap operations. The implementations of the operations are the same as for Fibonacci heaps, except that insertion and union consolidate the root list as their last step. What are the worst-case running times of operations on McGee heaps?

EXERCISE 4. Start to discuss how to approach the implementation of Fibonacci heaps in Computing Assignment 1. Prepare a time schedule how to approach the implementation of all tasks.