University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
P. Bamberger, P. Schneider

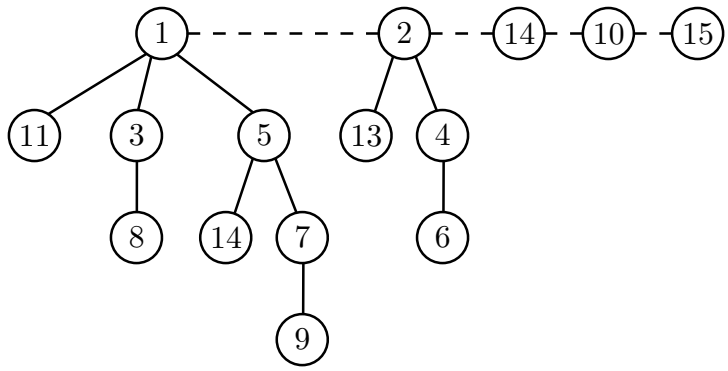
Algorithms Theory
Exercise Sheet 6

Due: Tuesday, 15th of December 2020, 4 pm

Exercise 1: Fibonacci Heap Worst Cases *(7 Points)*

(a) Consider the Fibonacci heap given below. Perform a `delete-min` operation. Give the state *before and after* the `consolidate` operation. Conduct the `link` operations *exactly* in the order in which the algorithm given in the lecture does it (Chapter 5 Part IV Slide 16). *(4 Points)*

(b) Give a valid instance of a Fibonacci heap where `delete-min` has a *worst case* runtime of $\Omega(n)$ and explain why this is the case for that instance. *(3 Points)*



Exercise 2: Fibonacci Heap Modifications *(13 Points)*

(a) Assume that operation `decrease-key` never occurs. Show that in this case, the maximum rank $D(n)$ of a Fibonacci heap is at most $\lfloor \log_2(n) \rfloor$. *(5 Points)*

(b) We want to augment the Fibonacci heap data structure by adding an operation `increase-key`(v, k) to increase the key of a node v (given by a direct pointer) to the value k. The operation should have an amortized running time of $\mathrm{O}(\log n)$. Describe the operation `increase-key`(v, k) in sufficient detail and prove the correctness and amortized running time. *(8 Points)*

Remark: You can use the same potential function as for the standard Fibonacci heap data structure. Note however that after conducting `increase-key`(v, k) *the Fibonacci heap must still be a list of heaps, where the maximum rank* $D(n) \in O(\log n)$.