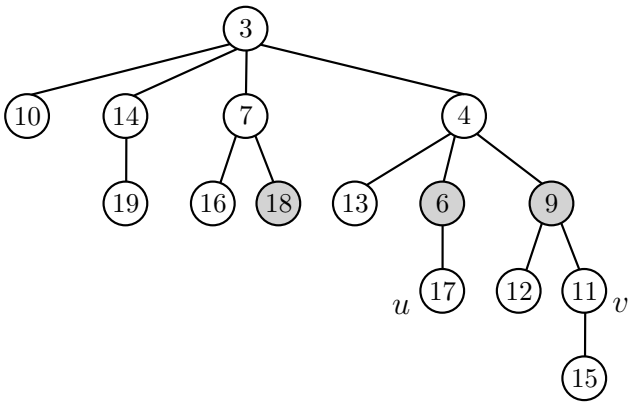
University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
P. Bamberger, P. Schneider

Algorithm Theory
Exercise Sheet 6

Due: Tuesday, 30th of November, 2021, 4 pm

Exercise 1: Fibonacci Heap - Operations *(4 Points)*

Consider the following Fibonacci heap with marked nodes shown in gray and two dedicated nodes u, v. Give the state of the Fibonacci heap after conducting the operation `Decrease-Key`$(v, 8)$. Then conduct `Decrease-Key`$(u, 5)$ on the resulting Fibonacci heap and give the state of it.



Exercise 2: Fibonacci Heap - Questions *(6 Points)*

Suppose we "simplify" Fibonacci heaps such that we do *not* mark any nodes that have lost a child and consequentially also do *not* cut marked parents of a node that needs to be cut out due to a `decrease-key`-operation. Is the *amortized* running time

(a) ... of the `decrease-key`-operation still $\mathcal{O}(1)$? *(2 Points)*

(b) ... of the `delete-min`-operation still $\mathcal{O}(\log n)$? *(4 Points)*

Explain your answers.

Exercise 3: Fibonacci Heap - Delete *(10 Points)*

We want to augment the Fibonacci heap data structure by adding an operation `delete(`v`)` to delete a node v (given by a direct pointer). The operation should have an amortized running time of $\mathcal{O}(\log n)$. Describe the operation `delete(`v`)` in sufficient detail and prove the correctness and amortized running time.

Remark: You can use the same potential function as for the standard Fibonacci heap data structure. Note however that after conducting `delete(`v`)` *the Fibonacci heap must still be a list of heaps with maximum rank $D(n) \in O(\log n)$ and with a dedicated pointer to the minimum key.*