University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
S. Faour, M. Fuchs, A. Malyusz

Algorithm Theory
Sample Solution Exercise Sheet 6

Due: Friday, 1st of December 2023, 10:00 am

Exercise 1: Fibonacci Heap Simulation *(10 Points)*

Do a consolidation operation on this heap then a delete-min.

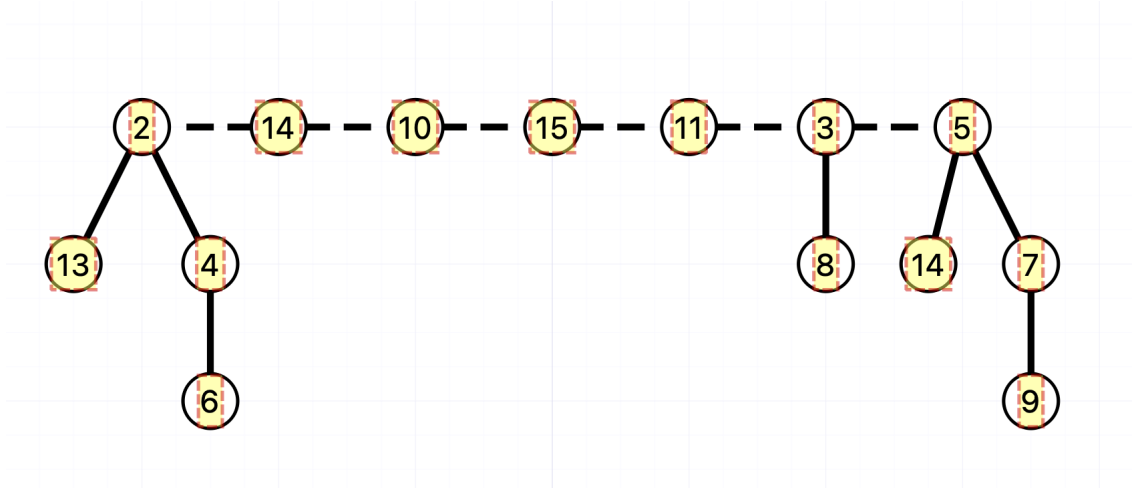

Figure 1: Before consolidation

Give the Fibonacci heap after the operations. Important: Follow the algorithm exactly as described on https://ac.informatik.uni-freiburg.de/teaching/ws23_24/algo_theo/pdf/Chapter5_PartIV.pdf.

Sample Solution

After consolidation:

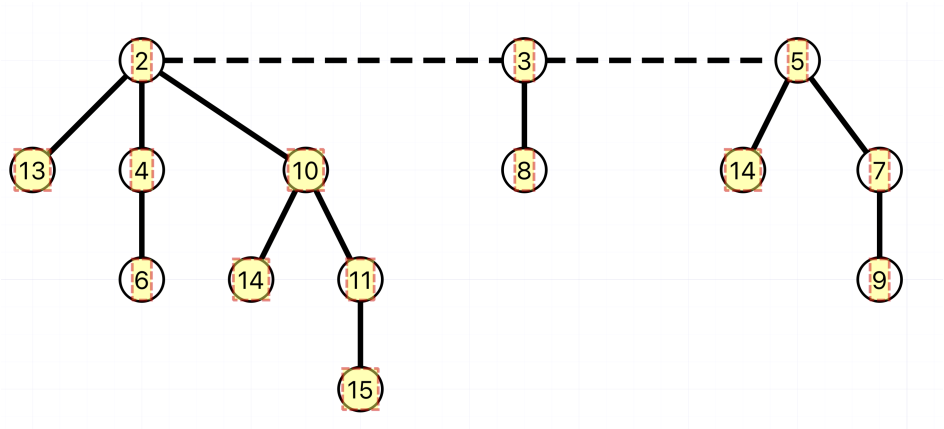

Figure 2: After Delete-Min

After Delete-Min:

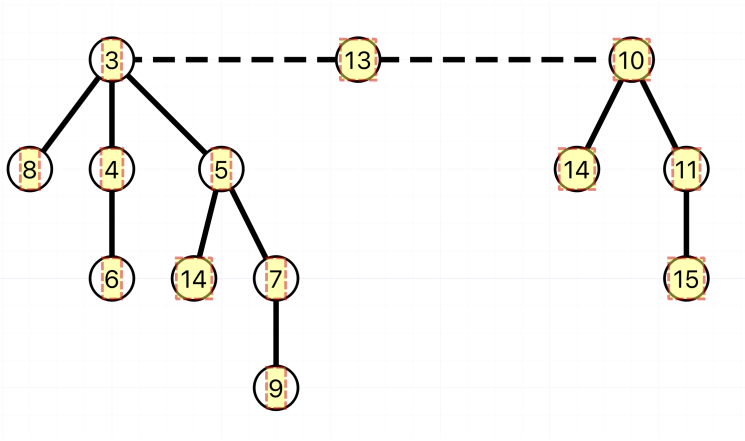

Figure 3: After consolidation

Exercise 2: Fibonacci Heap Properties *(3+7 Points)*

(a) Create a new method called Delete-node(v), which deletes node v from the Fibonacci heap in $O(\log n)$ amortized time.

(b) A crazy person at the bus stop claims that for every tree in a Fibonacci heap, the height is $O(\log n)$ (in other words: at most $c \log n$), where n is the number of nodes in the heap. Show that there is a Fibonacci heap that consists of only one tree, which is a chain (path) of n nodes.

Sample Solution

(a) Let your delete min function be the following.
FIB-HEAP-DELETE(H, x)
1. FIB-HEAP-DECREASE-KEY(H, x, $-\infty$)
2. FIB-HEAP-EXTRACT-MIN(H)
We know that the 1st operation is $O(\log n)$ amortized the 2nd operation $O(1)$ and like this we get the required runtime.

(b) You can easily create a chain of length 2 as such assume we already have a chain of length n. After this do the following steps iteratively. Add 3 nodes to the Heap (u, v, w). One of the elements of the 3 new nodes should have the smallest value (u) in the whole heap and another the 2nd smallest value (v). Use Delete-Min operation. During the consolidation the chain of length n will go under v thus creating a path of length $n + 1$. Now to truly get a chain just delete w from the heap.