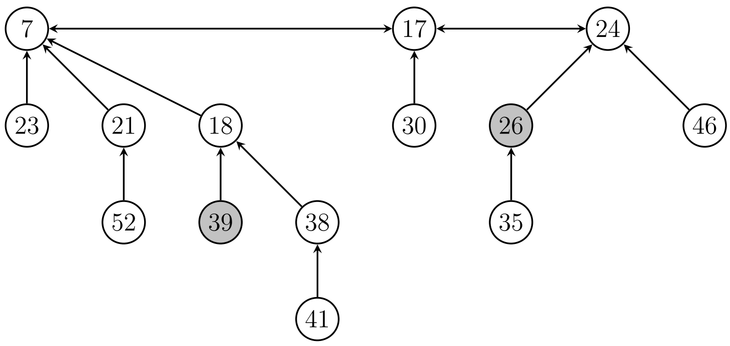
Universität Siegen
Lehrstuhl Theoretische Informatik
Markus Lohrey

Exercise 6

Task 1
Given the following Fibonacci heap:



Perform the following operations in that order:
delete-min, **decrease-key**(“52”, 9), **decrease-key**(“46”, 3), **insert**(42), **delete-min**, **decrease-key**(“35”, 7)

Task 2
Show Theorem 17 from the lecture: For all $k \geq 0$ we have

$$F_k = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^{k+1} - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^{k+1}.$$

On the last sheet we used $F_0 = 0$ and $F_1 = 1$, but here we use $F_0 = F_1 = 1$!

Task 3
Prove or disprove: The height of a Fibonacci heap of size n is at most $O(\log n)$.

Task 4
Find the optimal order to compute the following product (only the dimensions of the matrices are given):

$$(2 \times 4) \cdot (4 \times 6) \cdot (6 \times 1) \cdot (1 \times 10) \cdot (10 \times 10)$$

Task 5
Let $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots, y_n)$ be two sequences. We say X is a *subsequence* of Y if there are indices $1 \leq i_1 < i_2 < \cdots < i_m \leq n$ such that for all $1 \leq j \leq m$ it holds that $x_j = y_{i_j}$.
Use dynamic programming to implement an algorithm that runs in polynomial time which, given two sequences X and Y, computes the length of the longest common subsequence of X and Y.