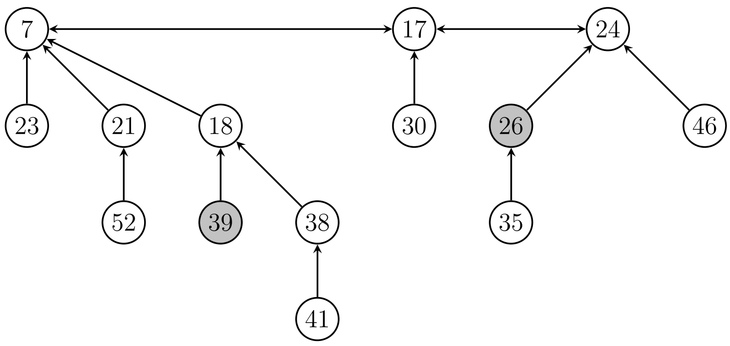
Universität Siegen
Lehrstuhl Theoretische Informatik
Markus Lohrey

Exercise 7

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
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 3
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 $1 \le i_1 < i_2 < \cdots < i_m \le n$ such that for all $1 \le j \le m$ 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.

Task 4
Construct an optimal binary search tree for the following elements v with probability (weight) $\gamma(v)$.

v	1	2	3	4	5	6
$\gamma(v)$	0.25	0.1	0.2	0.15	0.25	0.05

Task 5
Assume we want to construct an optimal binary search tree using the following greedy algorithm: Choose an element v for which $\gamma(v)$ is maximal as the root node and then continue recursively. Show that this approach does not always yield an optimal binary search tree.