1. Analyze the time and space complexity of the programme \texttt{compute}(k) (in \Theta-notation as a function of \(k\)), when \texttt{compute} is as follows.

\texttt{compute}(k)
\begin{center}
\begin{array}{l}
\text{if } k \leq 1 \\
\quad \text{then return} \\
\quad x \leftarrow 0 \\
\text{for } x \leftarrow 1 \text{ to } k \\
\quad \text{do } x \leftarrow x + 1 \\
\quad \text{print}(x) \\
\text{compute}(k/2)
\end{array}
\end{center}

Here \(k\) is a positive integer. The division on the last row of the programme is cut to an integer, e.g., \(7/2 = 3\). You can assume, for simplicity, that \(k\) is of the form \(2^n\) for some integer \(n\).

2. Implement a queue as a doubly linked circular list without sentinel: write detailed constant time algorithms for \texttt{enqueue}(Q, k), \texttt{dequeue}(Q) and \texttt{empty}(Q).

Note: Use links, not tables.

3. Show the most important steps, when the following elements are inserted one by one in this order to a min-heap originally empty: 5, 3, 17, 10, 24, 2 ja 19. Thereafter, delete the top element from the heap.

"Most important steps" mean the resulting tree after each insertion or deletion, as well as intermediate steps, in which something essential happens. You do not need to repeat those parts of the tree that do not change, as long as all changes are clearly shown.

4. You are given as input a table with \(n\) integers \(x_1, x_2, \ldots, x_n\). Give an algorithm working in time \(O(n \log n)\) that prints out the number \(\min\{ |x_i - x_j| \mid i \neq j \}\), i.e., the smallest difference between two different numbers in the table.

5. The \textit{longest increasing subsequence problem} is as follows: Given numbers \(a_1, a_2, \ldots, a_n\), find the maximum value \(k\) such that \(a_{i_1} < a_{i_2} < \cdots < a_{i_k}\) and \(i_1 < i_2 < \cdots < i_k\). As an example, if the input is 3, 1, 4, 1, 5, 9, 2, 6, 5, the maximum increasing subsequence has length four (1, 4, 5, 9 among others). Give a detailed algorithm, working in time \(O(n^2)\), that solves the problem.