# 2nd tutorial



**1. KMP.** Construct the search automata for the words "kokos" and "ananas". You can first just do it by hand from the definition, but then also go back to the KMP algorithm for the construction of the automaton and verify that it indeed works.

**2. Rotation.** A rotation of a string $\alpha$ is the string $\alpha[K:]\alpha[:K]$ for some $K, 0 \leq K < |\alpha|$, e.g. "nasana" is a rotation of "ananas" for $K = 3$.
Design an algorithm which decides whether $\alpha'$ is a rotation of $\alpha$.

**3. Periodicity.** How to decide whether a word $\alpha$ is *periodic*? By this we mean that there exists a word $\beta$ and a number $k > 1$ s.t. $\alpha = \beta^k$, that is, the concatenation of $k$ copies of $\beta$.

**4. Ordered trees.** Say that a tree is *ordered* if it is rooted and for each node, there is an order on its children (i.e., we can speak of the first, second, third etc. child). A subtree of an ordered tree is obtained by choosing a node, taking its subtree, and then possibly also removing some edges coming from the root, but only from the left and the right, that is, the edges which remain have to form a contiguous sequence of children (e.g., if the chosen node as 5 children, we can remove the first, fourth, and fifth, but not the first, third, and fifth).
Given two ordered trees, decide whether one is a subtree of another.

**5. Counting words.** You are given a string $S$ of length $k$ and a number $n$. How to efficiently compute the number of strings of length $n$ over the alphabet $\{\texttt{a}, \ldots, \texttt{z}\}$ which do not contain $S$ as a substring?

**6. Fibonacci words.** Let us define Fibonacci words as follows: $F_0 = \texttt{a}$, $F_1 = \texttt{b}$, $F_{n+2} = F_n F_{n+1}$. Design an algorithm which, in a given string over the alphabet $\{\texttt{a}, \texttt{b}\}$ finds the longest Fibonacci subword.

**7. Minimum rotation:** Design a linear algorithm which finds the lexicographically minimal rotation of a string $\alpha$.