Topics for presentations — winter term 2021/22

Topic 1 (Knuth-Morris-Pratt)
Explain in detail the Knuth-Morris-Pratt algorithm (based on finite automata) for searching of a pattern in a text and illustrate it on an example.
Analyze in detail the computational complexity of the algorithm.
As a source you can use the article:

> Donald E. Knuth, James H. Morris, Vaughan R. Pratt. Fast Pattern Matching in Strings. *SIAM Journal on Computing*, Vol. 6, No. 2, pp. 323–359, June 1977.

(*Remark:* The given text could be found in an electronic form also in MS Teams in the team for the course Theoretical Computer Science.)

Topic 2 (Greibach normal form)
A context-free grammar $\mathcal{G} = (\Pi, \Sigma, S, P)$ is in the *Greibach normal form* if all rules in P are of the form $A \longrightarrow a\alpha$, where $A \in \Pi$, $a \in \Sigma$, and $\alpha \in \Pi^*$. Moreover, as special rule $S \longrightarrow \varepsilon$ is allowed as an exception. However, if the grammar contains this rule then the nonterminal S can not occur on the right-hand side of any rule.
Describe in detail a general algorithm that transforms a given arbitrary context-free grammar to Greibach normal form, where moreover it will hold for every rule $A \longrightarrow a\alpha$ that $|\alpha| \leq 2$.

Topic 3 (Two-way pushdown automaton)
Construct a two-way deterministic pushdown automaton recognizing the language $\{\, ww \mid w \in \{\mathtt{a}, \mathtt{b}\}^*\}$.

Remark: The fact that the given finite automaton is two-way means that the head on the input tape can move in both directions and that this input word is surrounded from both sides with endmarks $\vdash$ and $\dashv$.

Topic 4 (Quicksort)
Give a detailed proof that the time complexity of the Quicksort algorithm for sorting in an average case is $\Theta(n \log n)$.

Topic 5 (The lower bound for the sorting)
Show that every algorithm that solve the problem of sorting (where the goal is to sort a given array of size n) that is based on comparing elements has necessarily a time complexity in $\Omega(n \log n)$.
By an algorithm based on comparing elements we mean that the only operations that the algorithm can do with values of elements of the array is to copy them (e.g., in the given

array from one index to another) and to compare them (i.e., to find out whether it holds for given values x and y that $x < y$, $x \leq y$, etc.).

Remark: Note that most of sorting algorithms such as Bubble sort, insertion sort, selection sort, Mergesort, Heapsort, Quicksort, etc., is based on comparing of elements.

Topic 6 (Decidability and undecidability)
Consider the problem

NAME: UHP (*Uniform Halting Problem*)

INPUT: Turing machine M.

QUESTION: Does M halt on every input?

Find out whether this problem is decidable or undecidable and prove your answer. In the case that the problem is decidable, show an algorithm that solves this problem; in the case that the problem is undecidable, you can start with a known undecidability of halting problem, and to show the corresponding reduction.