## Lecture

We continue with string matching. Last time we studied the string matching problem: given a pattern $P$ and a text $T$ check if $P$ is a substring of $T$. We saw two different algorithms to solve the problem: the string matching automaton and the KMP algorithm (Knuth-Morris-Pratt).

At the lecture today we will talk about string *indexing*. Here we want to build a data structure over a text $T$, such that when a query comes in form of pattern $P$, we can efficiently check wether the pattern is a substring of the text. The data structures we will use are *tries* and *suffix trees*.

You should read Chapter 12.3 in "Data Structures & Algorithms in Java" by Goodrich and Tamassia (available on CampusNet).

---

## Programming competition and mandatory assignments

The programming competition is now on. There will be a prize for the best three teams. You can find the rules and the description of the programming competition on the webpage.
Remember that you need to pass 3 of the mandatory assignments in order to be allowed to participate in the exam and pass two CodeJudge exercises (the programming competition can count as one fo these).

---

## Exercises

---

**Old exam set** Solve the exam set from E14 except exercise 6.