Assignment 02

Algorithms for Sequence Analysis

Sven Rahmann and Jens Zentgraf

21.04.2025, due **28.04.2025**

02.1 Aho Corasick (6 Theory)

Given the set of patterns $P = \{\texttt{aho, house, sea, rhesus, emu, user, us, use}\}$,

- build the Aho Corasick trie;
- extend the trie with the lps links
  (you may omit the links to the root, but only those);
- annotate each node with its output
  (although in practice you would use compressed lps links).

02.2: Recognizing cyclic permutations (4 Theory)

Given two strings s, t of the same length n,
how can you efficiently decide if one is a cyclic permutation of the other?
Def.: s is a cyclic permutation of t, if `s == t[i:] + t[:i]` for some i.

For example, `0123456` and `2345601` are cyclic permutations of each other,
but `6543210` is not a cyclic permutation of the other two (it's a reversal).
Also, `babbababbba` and `ababbbababb` are cyclic permutations of each other.

Tasks

1. Show that "is a cyclic permutation of" is an equivalence relation on strings.
2. Give an algorithm that takes two strings as input and outputs `True` or `False`.
   Your algorithm should run in $O(n)$ time.

02.3: BNDM (4 Theory)

For this task, $\Sigma := \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, $P := \mathtt{AGATATAGATCAG}$ and $T := \mathtt{AAGTAAAGTAGAGATAGATATAGATTAGCGT}$.

Illustrate the position of search windows when BNDM is used to search for P in T. Show how far each window is read, and how far the window is shifted in each step.

02.4: Shift-Or Algorithm (4 Programming)

- Implement the Shift-Or algorithm.
  You can adapt the code for the Shift-And algorithm from the slides.
- Make sure it runs correctly by running some tests,
  e.g. the example pattern and text from the lecture.