UNIVERSITÄT DES SAARLANDES

ZBI ZENTRUM FÜR BIOINFORMATIK

Assignment 03

Algorithms for Sequence Analysis

Sven Rahmann, Jens Zentgraf and Johanna Schmitz

28.04.2025, due **05.05.2025**

03.1 Matching Statistics (6 Theory)

Let s, t be strings with $|s| \leq |t|$.
Assume we have a suffix tree of s (the shorter string) **with suffix links**.

Let $|t| = n$. Let $M = M[0 : n]$ be an array of integers ("**matching statistics**"), such that $M[i]$ is the length of a longest substring that starts at position i in t, and that occurs also (somewhere) in s.

a. Describe how to compute M in $O(n)$ time, under the conditions stated above.
b. Describe how to obtain the longest common substring of s and t in $O(|s| + |t|)$ total time, given only s, t, with the help of M.

03.2 Suffix Tree (and Array) Example (5 Theory)

We use the alphabet order $\$ < \mathtt{A} < \mathtt{T}$.

[4T] Construct the suffix tree for the string $s = \mathtt{AATTAATT\$}$ by Ukkonen's algorithm.
Label each leaf with the starting position of the corresponding suffix of s.
Show the tree after each phase.
Give a textual description of what happens in each phase.
Take care to order the children of each node alphabetically.

[5T] What is the suffix array of s?

[0T] Make sure you also understand how Ukkonen's algorithm works on $\mathtt{AAAAA...\$}$ and $\mathtt{ABCDEFG...\$}$, assuming the usual alphabet order.

03.3 Naive suffix tree construction (2 Theory)

Consider using an optimal comparison-based sorting algorithm on the suffixes (mergesort) to build the suffix array of $T\$$.
Assume that $|T\$| = n$ and that r is the length of the longest repeated substring(s) in T.

What is the best bound on the worst-case running time of suffix sorting with this method and the given information?

03.4 Maximal Unique Matches (4 Theory)

Definitions

- Let two strings $s, t \in \Sigma^*$ be given.
- A string u is a **unique match** if it occurs **exactly** once in both s and t, respectively.
- A unique match u is **maximal** if there is no $a \in \Sigma$, such that au or ua is a unique match.

Given a suffix tree of $s\#t\$$, describe a linear-time algorithm for finding all MUMs.

Hints

Take the longest common substring algorithm as a starting point.
The algorithm might be split into two phases: tree annotation and tree traversal.