Technische Universität München
Department of Informatics
Chair for Efficient Algorithms
Prof. Dr. Ernst W. Mayr/Dr. Jens Ernst
Johannes Nowak

Summer Term 2005
Exercise Sheet 5
May 31, 2005

Selected Topics in Computational Biology

Due: 06.06.2005 **after** *the lecture*

Exercise 1 (10 points)

Let $t \in \Sigma^*$ and $k \in \mathbb{N}$. Design a linear-time algorithm that finds the shortest substring of t that occurs exactly k times in t.

Exercise 2 (10 points)

Determine the alignment distance of the strings $u = abcbcbabcaaabc$ and $v = bcbabbabacacbb$. The cost function $\overline{w} : \overline{\Sigma} \times \overline{\Sigma} \rightarrow \mathbb{R}_0^+$ is defined by

$$\overline{w}(x, y) = \begin{cases} 0 & \text{if} \quad x = y \neq \{-\} \\ 1 & \text{else} \end{cases}$$

Give also an optimal alignment of the strings. Is the optimal alignment unique?

Programming Task

Implement a program that solves the dictionary matching problem:
Given a set of patterns $D = \{p^1, p^2, \ldots, p^s\}$ over alphabet Σ such that $d = \sum_{i=1}^{s} |p^i|$ is the sum of the lengths of all patterns. A query takes an arbitrary string $t \in \Sigma^*$ of length n as input and reports all positions in the text where some pattern p^i starts inside t. Preprocess D in time $O(d)$ such that queries can be answered in time $O(n + tocc)$, where $tocc$ is the number of such occurrences.

- You can use the solution with suffix trees presented in the lecture.
- Your implementation should be in plain ANSI C.
- While it is essential to respect the asymptotic bounds given above, your solution will be also graded according to the involved constants with respect to both time and space. Think about what information is needed to represent the necessary data structures.
- You can test your program using the file *dictionaryinput.txt* which can be downloaded from the exercises webpage.