TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

Fachbereich Informatik
AG Algorithmen und Komplexität

Issue Date: 24.11.2014
Version: 2014-11-25 18:02

4th Exercise Sheet for Kombinatorische Algorithmen, WS 14/15

Hand In: *Until Monday, 01.12.2014, 12:00,*
deliver or email to Raphael (`reitzig@cs.uni-kl.de`).

Whenever an exercise states, "Develop an algorithm..." (or similar), your presentation should include a precise description of the algorithm as well as arguments of correctness and an analysis of its (relevant) costs (such as runtime, memory usage, ...).

Problem 6

4 + 2 points

This exercise will lead you towards an efficient (w. r. t. runtime) implementation of the LZ77-decomposition as defined in Problem 4.

Note that practitioners will want to use constrained versions which get by with a constant amount of memory for the price of worse compression rates.

a) As a first step, consider the following problem:

Longest Prefix Matching

Input: Text $T \in \Sigma^n$, pattern $P \in \Sigma^m$ and index $t \in [1..n]$.

Output: Length $\ell_{\max}$ of the longest prefix of P which occurs in T "before" position t and its matching site, i. e.

$$\ell_{\max} \;:=\; \max\Big\{\ell \in [0..\min\{n,m\}] \;\Big|\; \exists i \in [1..n-\ell] : i \leq t \wedge T_{i,i+\ell-1} = P_{1,\ell}\Big\}$$

and arbitrary

$$j \in \left\{i \in [1..n-\ell_{\max}] : i \leq t \wedge T_{i,i+\ell_{\max}-1} = P_{1,\ell_{\max}}\right\} .$$

Develop an algorithm that solves the Longest Prefix Matching problem in time $\mathcal{O}(t + \ell_{\max})$ using $\mathcal{O}(\ell_{\max})$ memory[1]. Less efficient algorithms may yield partial credit.

Hint: Some algorithms we discussed at the beginning of the course may be a good starting point.

[1] Memory constraints are always meant in addition to the input.

b) Develop an algorithm which computes LZ77(w) for $w \in \Sigma^n$ in time $\mathcal{O}(Cn)$ with $C := |\text{LZ77}(w)|$ the number of phrases in the LZ77-decomposition of w.

You may use an algorithm as specified in a) as subroutine (even if you did not come up with your own solution).

Maybe I should have compressed the problem statements more.