58093 String Processing Algorithms
Separate Exam, 14 August 2012 Examiner: Juha Kärkkäinen

Please write on each sheet: your name, student number or identity number, signature, course name, exam date and sheet number. You can answer in English, Finnish or Swedish.

1. [*4+4+4 points*] Each of the following pairs of concepts are somehow connected. Describe the main connecting factors or commonalities as well as the main separating factors or differences.

   (a) Shift–And algorithm and BNDM algorithm.

   (b) (Knuth–)Morris–Pratt algorithm and Aho–Corasick algorithm.

   (c) String quicksort and MSD radix sort.

   A few lines for each part is sufficient.

2. [*12 points*] A string A is a *subsequence* of a string B if A can be obtained by deleting characters from B. For example, `abc` is a subsequence of `abadc` but it is not a subsequence of `acadb`.

   Let P be a pattern and T a text. Describe a algorithm for finding the length of the shortest factor of T that contains P as a subsequence. For example, if $P =$ `abc` and $T =$ `cabadcabbddc`, then the answer is 5 as `abc` is a subsequence of $X =$ `abadc`, and X is shortest of such substrings of T. What is the time complexity of your algorithm in terms of the lengths of P and T?

3. [*4+4+4 points*] Give

   (a) the compact trie

   (b) the balanced ternary tree

   (c) the LLCP and RLCP arrays for efficient binary searching in the sorted array

   for the string set {`australia`, `austria`, `latvia`, `liberia`, `libya`, `lithuania`, `peru`, `somalia`, `spain`, `sudan`, `sweden`}.

4. [*12 points*] Define the suffix link in suffix trees and describe briefly its role in a linear time suffix tree construction algorithm.

5. [*12 points*] The task is to find the longest string S that occurs at least three times in a text T of length n. Describe how to find S in linear time given the suffix array of T and the associated LCP array without constructing any major additional data structures.