58093 String Processing Algorithms (Autumn 2012)
Renewal/separate Exam, 8 February 2013 at 16-20 Lecturer: 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. [*3+3+3 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) (Knuth–)Morris–Pratt algorithm and Shift-And algorithm.

   (b) String quicksort and string mergesort.

   (c) Aho–Corasick automaton and suffix tree.

   A few lines for each part is sufficient.

2. [*6+6 points*] Let $A = a_1 a_2 \cdots a_m$ ja $B = b_1 b_2 \cdots b_n$ two strings over the alphabet of real numbers, i.e., $a_i, b_j \in \mathbb{R}$ for all $1 \leq i \leq m$, $1 \leq j \leq n$. Let us define a variant of edit distance for such strings. The edit operations are the standard insertion, deletion and substitution of single symbols. The cost of substituting a_i with b_j is $|a_i - b_j|$, i.e., the absolute value of the difference. There are two models for the cost of insertions and deletions (indels):

   (a) The cost of inserting or deleting a symbol c is $|c|$, i.e., the absolute value of the symbol.

   (b) Indels have no cost, but the total number of indels must be at most K.

   Describe algorithms for computing these edit distance variants. The time complexity should be $\mathcal{O}(mn)$ for (a)-part and $\mathcal{O}(mnK)$ for (b)-part. You may assume that all basic arithmetic operations on real numbers can be performed in constant time.

3. [*3+3+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`, `libanon`, `libya`, `lithuania`, `mexico`, `singapore`, `spain`, `sudan`, `sweden`}.

4. [*9 points*] Define the suffix link in suffix trees and describe briefly its role in a linear time suffix tree construction algorithm.

5. [*10 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.