58093 String Processing Algorithms (Autumn 2012)
Course Exam, 13 December 2012 at 16-19 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. [*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) (Knuth–)Morris–Pratt algorithm and Aho–Corasick algorithm.

   (b) String quicksort and MSD radix sort.

   (c) Compact trie and suffix tree.

   A few lines for each part is sufficient.

2. [*7+7 points*] Let $T[0..n)$ be a string and let $lcp(T_i, T_j)$ denote the length of the longest common prefix between the suffixes of T starting at positions i and j. The *longest previous factor* array $LPF[1..n)$ is defined by

$$LPF[i] = \max_{j \in [0..i)} lcp(T_i, T_j) \ .$$

   (a) Show that for all $i \in [1..n-1)$, $LPF[i+1] \geq LPF[i] - 1$.
   Hint: If $S[0..p)$ is a prefix of T_i then $S[1..p)$ is a prefix of T_{i+1}.

   (b) Suppose we are given an array $Prev[1..n)$ of integers in $[0..n)$ satisfying for all i

$$Prev[i] < i$$
$$lcp(T_i, T_{Prev[i]}) = LPF[i]$$

   Describe an algorithm for computing the LPF array from the $Prev$ array in linear time. *Hint:* Use the result of (a)-part.

3. [*6+6 points*]

   (a) Compute the edit distance between strings `tukholma` and `stockholm` using the dynamic programming algorithm described on the course.

   (b) Give *all* optimal alignments between `tukholma` and `stockholm`, i.e., alignments with the same cost as the edit distance.

4. [*12 points*] Let T be a string of length n over an alphabet Σ of constant size. Describe an algorithm that finds the *shortest* string over the alphabet Σ that does *not* occur in T. The time complexity should be $\mathcal{O}(n)$.