58093 String Processing Algorithms

Renewal/separate Exam, 4 February 2011 at 16-20 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. [*3+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) Shift–Or algorithm and Myers' bitparallel algorithm.

   (b) Aho–Corasick algorithm and suffix tree.

   (c) String quicksort and MSD radix sort.

   (d) LCA (Lowest Common Ancestor) preprocessing ja RMQ (Range Minimum Query) preprocessing.

   A few lines for each part is sufficient.

2. [*3+2+3 points*]

   (a) Explain what are ordered alphabet and integer alphabet.

   (b) Give an example of an algorithm that works equally well with both kinds of alphabets.

   (c) Give an example of an algorithm that works well with one type of alphabet but not the other. Explain why the algorithm requires a specific type of alphabet.

3. [*3+3+3 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. [*11 points*] Let $\mathcal{R} = \{S_1, S_2, \ldots, S_k\}$ be a set of strings. String S_i and S_j are rotations of each other if $S_i = uv$ and $S_j = vu$ for some strings u and v. Describe an algorithm for finding all strings in $\mathcal{R}$ that are rotations of another string in $\mathcal{R}$. The algorithm should report each string only once even if it is a rotation of many other strings. The time complexity should be linear on a constant size alphabet.

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.