58093 String Processing Algorithms
Renewal/separate Exam, 3 February 2012 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. [*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) Aho–Corasick algorithm and suffix tree.

   (b) LSD radix sort and MSD radix sort.

   (c) Prefix doubling and induced sorting.

   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 exact string matching algorithm that works equally well with both kinds of alphabets.

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

3. [*10 points*] Use Ukkonen's cut-off algorithm to find all approximate occurrences of the pattern $P = \texttt{levee}$ in the text $T = \texttt{elevated\_water\_level}$ with edit distance $k = 1$.

4. [*10 points*] Let T be a string and let R be a multiset of symbols. A factor S of T is an occurrence of R if S consists of exactly the symbols of R. For example, if $T = \texttt{abahgcabah}$ and $R = \{\texttt{a}, \texttt{a}, \texttt{b}, \texttt{c}\}$, the only occurrence of R in T is the factor $S = \{caba\}$. Describe an algorithm for finding all occurrences of R in T. The time complexity should be $\mathcal{O}(|T|+|R|)$ on an alphabet of constant size.

5. [*10 points*] Let $\mathcal{R} = \{S_1, S_2, \ldots, S_k\}$ be a set of strings, where no string is a factor of another string. The *shortest distinguishing factor* of S_i is the shortest string that occurs in S but not in any other string in $\mathcal{R}$. Describe an algorithm for finding the shortest distinguishing factor for all strings in $\mathcal{R}$. The time complexity should be linear on a constant size alphabet.