58093 String Processing Algorithms
Separate Exam, 3 April 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) Horspool algorithm and BNDM algorithm.

   (b) String quicksort and ternary trie.

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

   A few lines for each part is sufficient.

2. [*4+8 points*]

   (a) Define the concept *prefix free*.

   (b) Explain with examples the role and significance of the concept *prefix free* in stringology.

3. [*12 points*] Construct the Aho–Corasick automaton for the pattern set {`angel`, `angry`, `chapel`, `gel`, `michael`}. Simulate the scanning of the text `michelangelo` with the automaton.

4. [*6+6 points*] Consider a variant of the edit distance that allows an unlimited number of *insertions* at the end of the string without a cost. In other words, the variant edit distance is

$$ed'(A, B) = \min\{ed(A, C) \mid C \text{ is a prefix of } B\} ,$$

   where $ed(\cdot, \cdot)$ is the standard edit distance.

   (a) Describe an algorithm that, given strings A and B, computes $ed'(A, B)$.

   (b) Describe an algorithm that, given strings A and B and an integer k, finds out whether B has a *suffix* B' such that $ed'(A, B') \leq k$.

   The time complexity should be $\mathcal{O}(|A||B|)$ in both cases. You may assume that any algorithms described on the lectures are known but any modifications to them should be described precisely.

5. [*12 points*] Let S and T be strings over the integer alphabet $[0..\sigma)$. Describe an algorithm that finds the shortest string that occurs in S but does not occur in T in $\mathcal{O}(|S| + |T| + \sigma)$ time. Justify the time complexity.