58093 String Processing Algorithms (Autumn 2014)
Renewal/Separate Exam, 06 February 2015 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. [*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 few lines for each part is sufficient.
   (a) Shift–And algorithm and Myers' bitparallel algorithm.
   (b) String quicksort and string mergesort.
   (c) Prefix doubling and induced sorting.

2. [*4+4+4 points*] Give for the string set {`australia`, `austria`, `latvia`, `libanon`, `libya`, `lithuania`, `mexico`, `singapore`, `spain`, `sudan`, `sweden`}
   (a) the compact trie
   (b) the balanced ternary trie
   (c) the LLCP and RLCP arrays for efficient binary searching in the sorted array

3. [*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. Formally, 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.

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 $\mathcal{O}(n)$.

Answer only one of the questions 5S and 5R. Answer 5S if you do NOT have the exercise and study group participation required for a renewal exam.

5S. [*12 points*] (*Separate exam.*) Describe any exact string matching algorithm covered in the study groups except Shift-And. Try to answer the following questions:
- What are the main ideas of the algorithm?
- How is the algorithm related to the algoriths described on the lectures?
- What kind of inputs the algorithm is particularly good on and why?

5R. [*12 points*] (*Renewal exam only.*) Any exact string matching algorithm can be used for multiple exact strings matching by searching each pattern separately, but some algorithms can be generalized to multiple patterns more efficiently. For example, the Aho–Corasick algorithm is a generalization of the Morris–Pratt algorithm. Describe such a generalization for some exact string matching algorithm other than (Knuth–)Morris–Pratt. You can choose any of the algorithms in the lectures but the asymptotic time complexity of your solution should be better than searching each pattern separately using the same algorithm.