

## RAM Algorithms 3

AVA Fall 2005, Final Exam

DIKU students, who follow the 5 ECTS version of the course should only hand in part 1 and 3, and not part 2 on "Dealing with hard problems".

1. **String matching.** Let  $T$  be a rooted tree with  $n$  nodes. An *ancestor* of a node  $v$  in  $T$  is the set of nodes on the path from the root to  $v$  (including  $v$  itself). The *common ancestors* of two nodes  $v$  and  $w$  are the set of nodes that are ancestors of both  $v$  and  $w$ . The *nearest common ancestor* of  $v$  and  $w$ , denoted  $nca(v, w)$ , is the common ancestor of  $v$  and  $w$  with maximum depth.

Suppose that there is a data structure for the tree  $T$ , that uses  $O(n)$  space and preprocessing time, and supports  $nca(v, w)$  for any two nodes  $v$  and  $w$  in  $O(1)$  time. Use this data structure to efficiently solve the following problem: Preprocess a string  $S$  of length  $n$  such that the following type of query can be handled efficiently:

**LCPrefix**( $i, j$ ): Return the length of the longest common prefix of the substrings  $S[i..n]$  and  $S[j..n]$ .

Give a data structure that supports **LCPrefix**( $i, j$ ) in  $O(1)$  time.

2. **Dealing with hard problems.** When solving the following problems, the on-line compendium on NP optimization problems (link available in the course schedule), will be useful (especially for problems (a) and (d)).
  - (a) Can NEAREST STRING be approximated within a factor 2 (if  $P \neq NP$ )? Give a reference to a relevant hardness or approximation result.
  - (b) Formulate a decision version of the NEAREST STRING problem and show that it is in NP.
  - (c) Does there exist a polynomial time reduction from the problem you defined in (b) to 3SAT? Motivate your answer.
  - (d) MINIMUM METRIC TRAVELING SALESPERSON PROBLEM is NP-Hard. Show that the problem is still NP-hard in the special case where all distances are 2 or 3.
3. **RAM Algorithms.** Given a sequence of  $n$  characters  $c_1, c_2 \dots c_n$  and their corresponding frequencies  $f_1, f_2, \dots f_n$  in a text, where each  $f_i$  is at most  $m$ . How fast can you construct a Huffman tree? (Give a high level description of your algorithm.)

Suppose the frequencies satisfy the following condition that, for each  $i$ ,  $2 \leq i \leq n-1$ : the sum of the least  $i$  frequencies (i.e., the first  $i$  frequencies in the non-decreasing order of the frequencies) in the sequence is less than the  $(i+1)$ -st least frequency (i.e., the  $(i+1)$ -st in the non-decreasing order of the frequencies). How fast can you construct a Huffman tree in this case? (Consider both cases:  $n \geq m$  and  $n < m$ .)