58093 String Processing Algorithms (Autumn 2014)
Exercises 2 (November 4)

Solve the following problems before the exercise session and be prepared to present your solutions at the session.

1. Outline algorithms that find the most frequent symbol in a given string

   (a) for ordered alphabet, and

   (b) for integer alphabet.

   The algorithms should be as fast as possible. What are their time complexities?

2. Let $\mathcal{R} = \{\texttt{manne}, \texttt{manu}, \texttt{minna}, \texttt{salla}, \texttt{saul}, \texttt{sauli}, \texttt{vihtori}\}$.

   (a) Give the compact trie of $\mathcal{R}$.

   (b) Give the balanced compact ternary trie of $\mathcal{R}$.

3. What is the time complexity of prefix queries for

   (a) trie with constant alphabet

   (b) compact trie with constant alphabet

   (c) compact trie with ordered alphabet and binary tree implementation of the child function

   (d) balanced compact ternary trie?

   The queries should return the resulting strings as a list of pointers or other identifiers rather than the full strings.

4. Show how to construct the compact trie for a set $\mathcal{R}$ in $\mathcal{O}(|\mathcal{R}|)$ time (rather than $\mathcal{O}(||\mathcal{R}||)$ time) given the string set $\mathcal{R}$ in lexicographical order and the LCP array $LCP_{\mathcal{R}}$.

5. Describe how to modify the LSD radix sort algorithm to handle strings of varying lengths. The time complexity should be the one given in Theorem 1.27.