58093 String Processing Algorithms (Autumn 2013)
Exercises 3 (12 November)

1. Show how to construct the compact trie $trie(\mathcal{R})$ 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 $LCP_{\mathcal{R}}$.

2. Use the lcp comparison technique to modify the standard insertion sort algorithm so that it sorts strings in $\mathcal{O}(\Sigma LCP(\mathcal{R}) + n^2)$ time.

3. Give an example showing that the worst case time complexity of string binary search without precomputed lcp information is $\Omega(m \log n)$.

4. Let $S[0..n)$ be a string over an integer alphabet. Show how to build a data structure in $\mathcal{O}(n)$ time and space so that afterwards the Karp–Rabin hash function $H(S[i..j))$ for the factor $S[i..j)$ can be computed in constant time for any $0 \le i \le j \le n$.

5. The Knuth–Morris–Pratt algorithm differs from the Morris–Pratt algorithm only in the failure function, which can be defined as

   $fail_{\text{KMP}}[i] = k$, where k is the length of the longest proper border of $P[0..i)$ *such that* $P[k] \neq P[i]$, or -1 if there is no such border.

   (a) Compute both failure functions for the pattern `ananassana`.

   (b) Give an example of a text, where some text character is compared three times by the MP algorithm but only once by the KMP algorithm when searching for `ananassana`.

6. Modify Algorithm 2.6 on the lecture notes to compute $fail_{\text{KMP}}$ instead of $fail_{\text{MP}}$.