58093 String Processing Algorithms (Autumn 2012)
Exercises 3 (15 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. Let R be a multiset containing n elements but only $d < n$ *distinct* elements. Show that ternary quicksort sorts R in $\mathcal{O}(n \log d)$ time. *Hint:* Sum up the maximum number of comparisons for each element and use the result in Exercise 2.6.

3. 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 1.23.

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

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

6. 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 $0 \leq i \leq j \leq n$.