58093 String Processing Algorithms (Autumn 2016)
Exercises 3 (Tuesday, November 15)

1. 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.

2. $\Omega(\Sigma LCP(\mathcal{R}))$ is a lower bound for string sorting for any algorithm in the simple string model, i.e., if characters can be accessed only one at a time. However, the packed string model allows accessing $\Theta(\log_\sigma n)$ characters at a time.

   Develop a version of MSD radix sort for the packed string model. What is the time complexity?

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. Define

$$MLCP[mid] = \max\{LLCP[mid], RLCP[mid]\}$$
$$D[mid] = \begin{cases} 0 & \text{if } MLCP[mid] = LLCP[mid] \\ 1 & \text{otherwise} \end{cases}$$

   Show that, if we store the arrays $MLCP$ and D instead of $LLCP$ and $RLCP$, we can compute $LLCP[mid]$ and $RLCP[mid]$ when needed during the string binary search.

5. 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 $0 \leq i \leq j \leq n$.