6332 - Advanced algorithms, Spring 2015, CSE, OSU
Homework 1
Instructor: Anastasios Sidiropoulos
Due date: Feb 4, 2014

Problem 1. The construction of suffix trees given in class assumes that the alphabet is of constant size, that is $|\Sigma| = O(1)$. Explain how Ukkonen's algorithm can be modified for the case where the alphabet size is not constant. More specifically, show how to construct a suffix tree for a string of length m in time $O(m \log |\Sigma|)$, and how to perform a query for a string of length n in time $O(n \log |\Sigma|)$.

Problem 2. Let Σ be an alphabet of constant size. Recall that the *Substring Problem for a Database* is as follows. The input consists of a set $\mathcal{T}$ of strings with alphabet Σ and of total length m. The goal is to preprocess $\mathcal{T}$ so that given a query string $P \in \Sigma^n$, we can find all occurrences of P in all strings in $\mathcal{T}$.

The solution we discussed in class for this problem was the following. Let $\mathcal{T} = \{T_1, \ldots, T_k\}$. We build a suffix tree for the string $S = T_1\$_1T_2\$_2 \ldots T_k\$_k$, where the symbols $\$_1, \ldots, \$_k$ denote distinct characters that are not in Σ. Using the solution of Problem 1 this idea can be implemented with alphabet $\Sigma' = \Sigma \cup \{\$_1, \ldots, \$_k\}$, which is of size $k + O(1)$. This results in preprocessing time $O(m \log k)$ and query time $O(n \log k)$.

Show how to solve this problem with preprocessing time $O(m)$ and query time $O(n)$.

Problem 3. Let Σ be an alphabet of constant size. A substring $P \in \Sigma^*$ is called a *prefix repeat* of a string $S \in \Sigma^*$ if P is a prefix of S and it is of the form $P = QQ$ for some string $Q \in \Sigma^*$. Give a linear-time algorithm to find the longest prefix repeat of an input string S.