TEL AVIV UNIVERSITY
Department of Computer Science
0368.4281 – Advanced topics in data structures
Fall Semester, 2006/2007

Homework 1, Oct 30, 2006

Due on Monday Nov 13. Don't forget to keep a copy of the homework.

1. A string s of length n is *periodic* if there is a string u of length $\leq n/2$ such that

$$s = u^k u' \ , \tag{1}$$

where k is an integer ≥ 2, u^k is the concatenation of k copies of u, and u' is a prefix of u. The smallest period of s is the shortest u for which Equation (1) holds. Suppose you are given a suffix tree of s together with an LCA data structure show how to use it to find the smallest period of S or declare that S is not periodic.

2. A *maximal repeat* of a string s is a substring α of s such that there exist $i < j$ where $s_i, \ldots, s_{i+|\alpha|-1} = s_j, \ldots, s_{j+|\alpha|-1} = \alpha$. Furthermore, if $i > 1$, then $s_{i-1} \neq s_{j-1}$, and if $j+\alpha-1 < n$, then $s_{i+\alpha} \neq s_{j+\alpha}$.
(a) Prove that s has at most n maximal repeats. (Try to find 2 proofs: one based on suffix trees, and a direct argument.)
(b) Give an algorithm to find all maximal repeats in s. Prove its correctness and analyze its running time.
(c) A *nonoverlapping maximal repeat* is one where there exist $i < j$ as before, and in addition $i + |\alpha| - 1 < j$. Give an algorithm that finds the longest nonoverlapping maximal repeat.

3. Given a string s, $|s| = n$, the suffix array, SA, of s, is a permutation of $\{1, 2, \ldots, n\}$ such that $SA[j] = i$ if and only if the suffix of s starting with the character i, $(i = 1, \ldots, n)$ is the j^{th} when we order the suffixes lexicographically. We add a special character \$ to each suffix which is smaller than any other character so that the lexicographic order of the suffixes is well defined.
(a) Given a permutation π of $1, 2, \ldots, n$, is there always a string s of length n such that π is the suffix array of s ? Prove your answer.
(b) Below are three suffix arrays. For each of these suffix arrays find a string s of length n, over the smallest possible alphabet Σ, such that the correponding array is a suffix array of s. Prove that there is no string s' over a smaller alphabet such that the suffix array is a suffix array of s'.

1)

n	n-1	$\cdots$	2	1

2)

1	2	$\cdots$	n-1	n

3) Assume n is even:

n	1	n-1	2	n-3	3	$\cdots$	$n/2+1$	$n/2$

4. The recurrence for the running time of the algorithm for computing a suffix array presented in class is $T(n) = T(2n/3) + O(n)$. Show how to modify the algorithm to give one whose recurrence is $T(n) = T(3n/7) + O(n)$.

5. Suppose we have a pattern P and a text T and we want to find for each position of T the length of the longest substring of P that matches there. Design a linear time algorithm for this problem.