TEL AVIV UNIVERSITY
Department of Computer Science
0368.4281 – Advanced topics in data structures
Fall Semester, 2007/2008

Homework 2, June 15, 2008

Due on Sunday June 29. Don't forget to keep a copy of the homework.

1. Consider the following special case of the union-find problem. The elements are the integers $1, 2, \ldots, n$. Initially each integer i is in a singleton set $\{i\}$. Then we perform a sequence of $m \geq n$ union and find operations. However, we restrict the unions so that at any time each set is an interval of consecutive integers. We impose this restriction by allowing to unite only consecutive intervals. We represent an interval by its largest integer, so $find(i)$ returns the largest interger in the set containing i. Describe a data structure that performs any sequence of m such operations in $O(m)$ time.
Hint: Partition the problem into small problems as we did when solving the LCA problem.

2. 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.

3. (a) Describe a linear time algorithm to count all distinct substrings of a string s.
(b) We call a substring α of a string s a *minimal unique* substring of s if α occurs only once in s and any proper prefix of *alpha* occurs more than once in s. Describe a linear time algorithm to enumerate all minimal unique substrings of an input string s.

4. 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.

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.