Przemysław Uznański March 14, 2017

Advanced Data Structures

Spring Semester 2017

Exercise Set 4

Exercise 1:
Given integer k, text $T[1 \,..\, n]$ and pattern $P[1 \,..\, m]$, we say that it *matches with k-mismatches* at position i, if $T[i \,..\, i+m-1]$ and P differ in at most k positions. Describe $\mathcal{O}(nk)$ algorithm for finding all k-mismatches aligments.
Hint: You can actually do it in $\mathcal{O}(k)$ time per aligment.

Exercise 2:
A palindrome is a word that is identical to its reverse: $v = v^R$. Describe $\mathcal{O}(n)$ algorithm for finding longest palindromic subword.
Hint: You can actually find in $\mathcal{O}(1)$ time the longest subword centered at given position.

Exercise 3:
Describe efficient algorithm for finding longest substring which appears at least k times in a given text.

Exercise 4:
A rotation of word $T[1 \,..\, n]$ is a word of form $T[i+1 \,..\, n]T[1 \,..\, i]$, for some i. Describe algorithm for finding *lexicographically smallest rotation*.
Question: Can you give two algorithms, either using suffix arrays or suffix trees?

Exercise 5:
Describe algorithm for computing number of different substrings of a given word.

Exercise 6: (*)
Given text T and its suffix array SA, describe how to recover its LCP array in $\mathcal{O}(n)$ operations, without recomputing SA and LCP from scratch using algorithm from lecture.
Hint: Kasai et al. *"Linear-Time Longest-Common-Prefix Computation in Suffix Arrays and Its Applications"* has short and clever solution.

Exercise 7: (*)
Given a string S, find all of its periodic prefixes. A string T is periodic if it is of the form $w^k w[1 \,..\, i]$ for some integer $k > 0$, integer i and word w.
Hint: Try to match string S with one of its suffixes.