Karl Bringmann, Erik Jan van Leeuwen **Winter 2016/2017**

Exercises for Algorithms and Data Structures

http://www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/winter16/algorithms-and-data-structures/

Exercise Sheet 5 Due: **28.11.2016**

The homework must be handed in on Monday before the lecture. You may collaborate with other students on finding the solutions for this problem set, but every student must hand in a writeup in their own words. We also expect you to state your collaborators and sources (books, papers, course notes, web pages, etc.) that you used to arrive at your solutions.

You need to collect at least 50% of all points on the first six exercise sheets, and at least 50% of all points on the remaining exercise sheets.

Whenever you are asked to design an algorithm in this exercise sheet you have to give a proof of its correctness as well as an asymptotic upper bound on its worst case running time.

Exercise 1 (*10 points*)

We are given two strings T and T'. Design a fast algorithm that determines if T is a cyclic rotation of T' (for example "RITHMSALGO" is a cyclic rotation of "ALGORITHMS").

Exercise 2 (*10 points*)

In the Word Break problem we are given a text t and a dictionary D (i.e., D is a set of strings), where $|t| = n$ and $\sum_{d \in D} |d| = m$. The goal is to decide whether t can be split into a sequence of words from D.

Design an $O(n^2 + m)$ worst case running time algorithm using tries and dynamic programming. You are not allowed to use suffix trees for this exercise.

Exercise 3 (*10 points*)

We are given a text t of length n consisting of letters from an alphabet Σ. Use the suffix tree data strcture you have seen in class to design $O(n)$ worst case running time algorithms for the following two problems:

a) (5 Points) Compute the longest substring that appears at least twice in t. For example, for $t =$"ABBABBA" the answer would be "ABBA".

b) (5 Points) Compute the shortest string formed by letters from Σ that is not a substring of t. For example, if $\Sigma = \{A, B\}$ and $t =$"ABBA", then the output should be "AA".

Exercise 4 (*10+3 points*)

You are given a $n \times n$ matrix M whose entries are all integers (not necessarily all positive). Your goal is to find a submatrix S formed by contiguous rows $\{i_1, i_1 + 1, \ldots, i_2\}$ and columns $\{j_1, j_1 + 1, \ldots, j_2\}$ such that the sum of all entries of S is maximal (among all such submatrices of M).

For example if $M = \begin{pmatrix} -1 & 0 & -4 & -1 \\ -1 & 3 & 1 & 0 \\ -2 & 2 & 4 & -4 \\ -1 & 0 & -3 & -2 \end{pmatrix}$, then $S = \begin{pmatrix} 3 & 1 \\ 2 & 4 \end{pmatrix}$ is the submatrix with the largest sum over all its entries.

a) (3 points) Give an $O(n^6)$ time algorithm for this problem.

b) (7 points) Give an $O(n^4)$ time algorithm for this problem.
Hint: Precompute the values $V[i', j'] = \sum_{i=1}^{i'} \sum_{j=1}^{j'} M[i, j]$.

c) (3 bonus points) Give an $O(n^3)$ time algorithm for this problem.