Data Structures and Algorithms **Spring-Summer 2011/12**

Assignment 2

Valeria Fionda, Mouna Kacimi,
Werner Nutt, Simon Razniewski

Asymptotic Complexity and Substring Matching

Instructions: Your assignment should represent your own effort. However, you are not expected to work alone. It is fine to discuss the exercises and try to find solutions together, but each student shall write down and submit his/her solutions separately. It is good academic standard to acknowledge collaborators, so if you worked together with other students, please list their names.
You must be prepared to present your solution at the lab. If you are not able to explain your solution, this will be considered as if you had not done your work at all.
You can write up your answers by hand (provided your handwriting is legible) or use a word processing system like Latex or Word. Experience shows that Word is in general difficult to use for this kind of task.
For a programming task, your solution must contain (i) an explanation of your solution to the problem, (ii) the Java code, in a form that we can run it, (iii) instructions how to run it. For all programming tasks, it is not allowed to use any external libraries ("import"), if not stated otherwise.
Please, include name and email address in your submission.

1. Comparison According to Asymptotic Complexity

Order the following functions according to their asymptotic complexity, from the function having the smaller asymptotic complexity to the function having the larger one (i.e., such that $f_1 = O(f_2)$; $f_2 = O(f_3)$; ...):

- $50 \cdot \log_2 n$
- $5 \cdot n + n^2 + 1$
- $\log_{10}^2 n$
- $2^n + 5$
- $3^{\sqrt{n}}$
- $5^3 \cdot n$

- $3 \cdot \log_{10} n$
- $(n+1)!$
- $4^{\log_2 n}$
- $\sqrt{n}$.

(10 Points)

2. Asymptotic Equalities

Prove or disprove the following statements:

a) $2n^n + 2^{n+1} = 2n^n + \Theta(2^n)$

b) $(n+a)^b = \Omega(n^b)$ for $a, b > 0$

c) $8e + e{\cdot}\lg v = O(e)$

(5 Points)

2. Asymptotic Puzzles

For each one of the following statements, write two functions f and g that satisfy the given condition.

a) $f(n) = O(g^2(n))$;

b) $f(n) = \Omega(f(n)g(n))$;

c) $f(n) = \Theta(g(n)) + \Omega(g^2(n)))$.

(5 Points)

4. Substring Matching

Write a substring matching algorithm that satisfies the following specification:

> Given two character strings, string A of length n and string B of length m, the algorithm returns 0 if B is *not* a substring of A, and returns the start position s of B in A if B is a substring of A.

For example, if A = “`assignment`” and B = “`sign`”, then the algorithm should return the number 3.
Make sure that your algorithm is valid for all special cases of input data.
Write a Java program implementing your algorithm.

(10 Points)

Submission: Tue, 27 March 2012, 8:30 am. Preferably by email.

If you want to submit a hand-written solution, hand it over to your lab tutor or in the lecture. However, all code has to be submitted in electronic form.