Stockholm University
Department of Mathematics
Lecturer: Marc Hellmuth
Tutors: Anna Lindeberg, Patricia Alexandra Ebert, David Sermoneta, Marcus Hultvall Hultin

1. Exercise "Data Structures and Algorithms (DA 4006)"

Problem 1: 10 points

Back to the roots: Provide a pseudocode for multiplying two arbitrary positive integers m and n that only uses the following basic instructions:

basic arithmetics	`inc`($variable$) replaces the value of $variable$ by the value $variable + 1$, i.e., `inc`($variable$) can be read as $variable := variable + 1$
data movement	assigning a value 0 to a variable `RETURN`
iteration	`WHILE` (condition) `DO` instruction
comparison	$<, >, \neq, =, \geq, \leq$

In other words, only `WHILE`-loops, comparisons and assignments similar to $i := 0$ or `inc`(i) are allowed. Since it is not allowed to assign values other than 0 to variables, assignments as $i := m$ or $i := j$ are forbidden.
Explain shortly the basic idea behind your algorithm.

Problem 2: 8+2=10

A string S is a sequence $a_0 \ldots a_p$ of $p \geq 0$ letters from the Swedish alphabet $\mathcal{A} =$ {a,b, ...,z, å, ä, ö}. We use $\prec$ to denote the conventional ordering of the alphabet, i.e., $a \prec b \prec c \prec \cdots \prec z \prec$ å $\prec$ ä $\prec$ ö.
Given two strings $S = a_0 \ldots a_p$ and $T = b_0 \ldots b_q$ we say that string S is *lexicographically less* than string T (in symbols $S \prec^* T$) if either

1. there exists an integer j, where $0 \leq j \leq \min\{p, q\}$, such that $a_i = b_i$ for all $i \in \{0, 1, \ldots, j-1\}$ and $a_j \prec b_j$, or
2. $p < q$ and $a_i = b_i$ for all $i \in \{0, 1, \ldots, p\}$

Consider `Insertion_Sort(A)` as provided on Slide 12 in the course-slides (*1-Fundamentals-Slides.pdf*). Suppose, however, that the input A is not a list of integers but a list of strings and replace in Line 4 of the algorithm "$A[i] > key$"by "$key \prec^* A[i]$".

(a) Apply this adjusted `Insertion_Sort` on the following list A of strings

$$A = (\texttt{apppe}, \texttt{apple}, \texttt{appl}, \texttt{apap}, \texttt{appap})$$

Provide the list A after each of the jth execution of the for-loop.

(b) Reorder the elements in A such that the resulting list A' serves as a worst-case example for this adjusted `Insertion_Sort`.

Problem 3: 6+4=10

Algorithms A and B spend exactly $T_A(n) = 0.1 \cdot n^2 \cdot log_{10}(n)$ and $T_B(n) = 2.5 \cdot n^2$ microseconds, respectively, for a problem of size n.

(a) Explain which algorithm you would recommend for general n. That is, explain which algorithm is better in the Big-O sense.

(b) If your problems are of the size $n \leq 10^9$, which algorithm will you recommend to use? Explain your decision.

Problem 4: 5+5=10

Given is the following function:

```
function(int n, int N)
    IF (n ≤ 1) THEN return N
    N := N + 1
    function(⌊n/3⌋, N)
```

(a) Call `function`$(100, 0)$ and provide for each single call of this function the values n and N

(b) Determine the runtime of `function(int` n, `int` N) in Big-O notation (best possible bound) assuming that basic operations can be done in constant time without using the Master theorem. Prove your result.

Problem 5: Bonus 4

There are 10 stacks of 10 identical-looking items. All of the items in one of these stacks are counterfeit. All items in the other stacks are unfeigned. Every unfeigned item weighs 10 grams, and every fake weighs 11 grams. You have a digital scale that can determine the exact weight of any number of items. Provide an algorithm that finds all fake items using using the minimum number of weighings. Explain you results.

Bonus problems can be used to earn a certain amount of extra points counted for the final exam.

Deadline: Thursday - April 2, 2026