Course Information

Course: Fundamental Computer Science I

Course Number: 320201

Time: Tuesday, 8.29-9.30, Friday,11.15-12.30

Place: Lecture Hall Research III

Instructor: Dr. Holger Kenn, Tel: 3112,
E-mail: h.kenn@iu-bremen.de

Web page: http://www.faculty.iu-bremen.de/course/FundCS1/

Problem Session: TBA

TAs:

- Christoph Burger-Scheidlin
- Aakash Jain
- Andreas Pfeil
- TBA

On the course website, lecture notes and problem sheets will be available for download.

The grading scheme will be
50 % final exam
30 % mid term exam
20 % problem sheets

Problem sheets will be handed out on friday and will be handed in again on friday.
Problem sheets can/should be solved in groups and can also be handed in by the groups with a maximum of 2 **(TWO)** people per group. For each student of a group, the name and the matriculation number must be on the results handed in to be counted for grading. The usual rules of academic integrity apply.

Occasionally, one member of the group will be asked to the blackboard to explain her/his solution to the class. Her/his capability of demonstrating and defending the solution of the group to the class will determine the group's grading for that problemsheet.

Problem sheet 1

Course Fundamental Computer Science, Dr. Holger Kenn
e-mail: h.kenn@iu-bremen.de, tel.:+49 421 200 3112

This problemsheet's solution is to be handed in Friday, September 12th *before the lecture*, either clearly readable on paper or as a *PDF* file via e-mail to h.kenn@iu-bremen.de.

1.) Comparison of running times
For each function $f(n)$ and time t in the following table, determine the largest size n of a problem that can be solved in time t assuming that the algorithm to solve the problem takes $f(n)$ microseconds.

	1 second	1 minute	1 hour	1 day	1 month	1 year	1 century
$\log_2(n)$							
$\sqrt{n}$							
n							
$n \log_2(n)$							
n^2							
n^3							
2^n							
$n!$							

(4p, -0.5p per error/missing, minimum 0p)

2.) Inversions
Let $A[1..n]$ be an array of n distinct numbers. if $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an inversion.

2.1) List the five inversions of the array $\{2, 3, 8, 6, 1\}$. (1p)

2.2) What array with elements from the set $\{1, 2, \ldots, n\}$ has the most inversions? How many does it have ? (1p)

2.3) What is the relationship between the running time of insertion sort and the number of inversions in the input array? Justify your answer.(2p)

2.4) Give an Algorithm that determines the number of inversions in any permutation on n elements in $\Theta(n \log_2 n)$ worst-case time. (Hint: Modify merge sort)(2p)