Theory exam in DD1352/DD2352 Algorithms (data structures) and complexity
2015-08-19 9.00-11.00

No aids. Only paper and pencil. Write your answers directly on this form.

Bonus points from the academic year 2014/2015 can be used at this exam. For grade E 13 points is needed for the course DD1352 and 12 points for DD2352. If you also pass the grade D question below you will get D, and if you *also* pass the grade C question you will get C.

Name: ..... *Social security number:* .....

1. (6 p) Are the following clauses true or false? Circle the correct answer! For every question, the correct answer will give you 1 point, and a *convincingly motivated* answer will give you 2 points.

   a) $2^n \in \Omega(n^2)$.

   true **false**

   Motivation:

   b) The expected time complexity of Random Quicksort is $O(n)$.

   true **false**

   Motivation:

   c) NP$\subseteq$PSPACE

   true **false**

   Motivation:

2. (4 p) A, B, C, D and E are decision problems. Suppose that B is NP-complete, and that there are known polynomial Karp reductions between the problems in the following way (where a reduction from A to B is shown by an arrow A$\rightarrow$B):

   $$\begin{array}{ccccc} A & \leftarrow & B & \leftrightarrow & C \\ & & \uparrow & & \uparrow \\ & & D & \leftarrow & E \end{array}$$

   What is known about the complexity for A, C, D and E? Write an X in the table below for everything that is certain, and write a ring for everything that is possible but not certain.

	lies in NP	is NP-complete	is NP-hard
A			
C			
D			
E			

3. (4 p; 2 p for each subquestion)
   a) Define the concept *Karp reduction*.

   b) Define the concept *undecidable problem*.

4. (Question for grade D, grading criterion: *explain how problems of high complexity can be handled*)
   Give two different suggestions for how to attack NP-hard decision problems that are not optimization problems in disguise (as usual supposing P≠NP).

   Suggestion 1:

   Suggestion 2:

5. (Question for grade C, grading criterion: *design simple heuristics*)
   The optimization problem *Minimum Subset Sum* takes as input a goal M and a list of positive integers $t_1, t_2, \ldots, t_n$. In this question we suppose that $t_1 \leq t_2 \leq \ldots \leq t_n$.

   A valid solution to the problem is a subset of the numbers in the list with sum greater than or equal to M. We look for a solution of minimum sum.

   Your task is to design a heuristic for Minimum Subset Sum that first constructs a valid solution and then uses local search to improve the solution.

   Describe the algorithm in text or pseudocode.

   MinSubsetSumHeuristic(M, $(t_1, t_2, \ldots, t_n)$) =

   // Construction:

   // Local search: