# Tik–106.4100 Design and Analysis of Algorithms, autumn 2008

## Exam, December 16th, 2008

Write the following clearly on top of each paper you submit: "T-106.4100 Design and Analysis of Algorithms, December 16th, 2008", your full name, student ID and study programme, and the total number of papers you submit.

1. a) (3p) Which of the following conjectures are correct and which are incorrect? Give a short justification for each answer (exact mathematical proofs are not required)!

   i. $5n^2 + 12n + 5 \in O(n^3)$

   ii. $3n^3 + 3n \log n + 8n \in \Omega(n^2)$

   iii. $8n \log n + 5 \in \Theta(n^2)$

   b) (3p) Explain what is the difference between amortized complexity and average case complexity.

2. a) (3p) Solve the following recurrence, when $n$ is a power of four. An exact answer is required (an answer in $\Theta$ or $O$ notation is not enough).

$$T(n) = \begin{cases} 1, & \text{when } n = 1 \\ 5T(n/4) + 2n & \text{when } n > 1 \end{cases}$$

   b) (3p) Make a good guess to solve the following recurrence and check your result by using induction ($c_1$ and $c_2$ are constants).

$$T(n) \leq \begin{cases} c_1, & \text{when } n = 1 \\ 3T(n/3) + c_2 n & \text{when } n > 1 \end{cases}$$

3. a) (2p) For which purposes are heaps (for example a binary heap, a binomial heap or a Fibonacci heap) used? Give an example of an algorithm which uses the most important heap operations. You do not have to explain the whole algorithm completely. It is enough to explain briefly the purpose of the algorithm and how the algorithm uses heap operations.

   b) (4p) Explain the main differences between binomial heaps and Fibonacci heaps. Which heap operations have different time complexities in these structures? Give the time complexities of those operations (in both binomial and Fibonacci heaps) and give short justifications for the complexity results (exact mathematical analysis is not required).

4. (6p) Design a dynamic programming algorithm for the *change-making problem*: given an amount $n$ and unlimited quantities of coins of each of the denominations $\{d_1, d_2, \ldots, d_m\}$, find the smallest number of coins that add up to $n$ or indicate that the problem does not have a solution. In addition of the total number of coins needed, your algorithm has also to find out the number of coins of each denomination needed. (Note that you should minimize the total number of coins needed, not the number of different denominations.)

   Do not write the code of the algorithm, but present the expressions needed by the dynamic programming and tell how and in which order the values of these expressions are calculated. Explain also how the number of coins of each denomination needed can be found out.

   If your algorithm is not based on dynamic programming, you cannot obtain full points from your solution.

   Hint: Check that your solution works properly if $n = 110$ and the denominations are $\{20, 50\}$.

5. (6p) Let $G = (V, E)$ be a directed graph. Write a pseudocode for an algorithm which determines if $G$ contains a cycle. Your algorithm should run in $O(|V| + |E|)$ time. Your algorithm does not have to list the vertices in the cycle. It is enough to determine if the graph contains at least one cycle. Hint: only back edges can cause cycles.

**Please, fill the course feedback form. The link is in the Noppa page of the course!**