Tik–106.4100 Design and Analysis of Algorithms, autumn 2009

Exam, February 6th, 2010

Write the following clearly on top of each paper you submit: "T-106.4100 Design and Analysis of Algorithms, February 6th, 2010", your full name, student ID and study programme, and the total number of papers you submit.

1. a) (3p) Solve the following recurrence, when n is a power of three. An exact answer is required (an answer in Θ or O notation is not enough).

$$T(n) = \begin{cases} 1, & \text{when } n = 1 \\ 2T(n/3) + 2n & \text{when } n > 1 \end{cases}$$

   b) (3p) Make a good guess to solve the following recurrence and check your result using induction (c_1 and c_2 are constants and n is a power of four).

$$T(n) \leq \begin{cases} c_1, & \text{when } n = 1 \\ 4T(n/4) + c_2 n & \text{when } n > 1 \end{cases}$$

2. a) (2 p) Consider disjoint-set forests supporting FIND-SET and UNION operations. Why is it desirable that the tree representing a certain set has as small height as possible? Explain the reason using a few sentences.

   b) (1 p) How can the operation FIND-SET be performed to keep the height of the tree small (there are several alternative ways, explain just one)? Give an example with a figure.

   c) (3 p) Explain how the operation EXTRACT-MIN is performed for binomial heaps and for Fibonacci heaps. What is the **worst case** time complexity of these operations? Give short justifications for your time complexity results.

3. a) (2p) Explain briefly the principles of sweeping.

   b) (2p) Explain briefly the principles of branch-and-bound technique.

   c) (1p) What kind of problems are suitable to be solved with branch-and-bound technique?

   d) (1p) What can be told about the time complexity of branch-and-bound algorithms?

4. a) (1p) A binary number x is stored in an array so that each element of the array contains one bit of x. We want to perform a sequence of increment operations. Each increment operation adds 1 to the value of the number. Design an algorithm which has the worst case time complexity (for one increment operation) $O(k)$ where k is the number of bits having value 1. The size of the array is not changed during the operations. (The binary number has a maximum value. If the maximum value has been reached, the next increment operation sets its value to 0.) Changing a or examining the value of one element in the array takes a constant time.

   b) (5p) What is the amortized complexity of the sequence of n increment operations, if initially $x = 0$? Use the accounting or potential method to justify your answer.

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.