Tik–106.4100 Design and Analysis of Algorithms, autumn 2005

Exam, February 6th, 2005

Write the following clearly on top of each paper you submit: "T-106.4100 Design and Analysis of Algorithms, February 6th, 2006", 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 two. An exact answer is required (an answer in Θ or O notation is not enough).

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

   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 two).

$$T(n) \leq \begin{cases} c_1, & \text{when } n = 1 \\ T(n/2) + c_2 & \text{when } n > 1 \end{cases}$$

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?

   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 dynamic programming.

   b) (4p) Explain briefly how the selection algorithm (finding the ith smallest element of the array containing n elements) based on divide-and-conquer technique works. (The text book presents two selection algorithms, explain the algorithm having better worst case performance.) What is the worst case time complexity of this algorithm (justification is not required)? How would the worst case complexity change if we selected the pivot differently: for each partition, the last element in the area to be split is selected as pivot (give a justification for the new complexity)?

4. (6p) If the size of the array to be sorted is n, the average-case running time of quicksort is $O(n \log n)$. How is this result obtained? Explain one way to derive the result. You do not have to remember all details of the mathematical analysis, but tell the principles. Include also some equations from the beginning of the analysis. The recommended length of the answer is 1–2 pages. You may present the analysis for a randomized version of quicksort (given in the text book).

5. (6p) Consider the problem of calculating the maximum flow in a flow network (graph).

   Suppose we have a flow network $G = (V, E)$ (G is a directed graph) in which each edge $(u, v) \in E$ has a nonnegative capacity $c(u, v)$ and a current flow $f(u, v)$. Assume that the residual network of G induced by the current flow f has already been calculated. The residual network is denoted by $G(f)$. Write a pseudocode for an algorithm which calculates one shortest augmenting path from node s to node t by using the residual network $G(f)$. In addition to the augmenting path, your algorithm must calculate the maximal increase of flow in this augmenting path. The augmenting path calculated does not have to be the best of all augmenting paths, but it must be as short as possible.

   The input of the algorithm are the residual network $G(f)$ and the nodes s and t. The output of the algorithm are the edges belonging to the augmenting path and the maximal increase of flow in this augmenting path.