TAMPERE UNIVERSITY OF TECHNOLOGY
Department of Mathematics

MAT-72006 **Advanced Algorithms and Data Structures** Examination
prof. Tapio Elomaa Dec. 9, 2014

Write your name and student number to each separate answer sheet. Neither calculators nor any other extra material is allowed in the exam.

Return the question sheet together with your answers. You are not allowed to exit the exam with the question sheet.

All students are required to answer questions 1 and 2. In addition, you may choose to answer any two questions from among 3–6. The maximum score for questions 1 and 2 is 8 points and for questions 3–6 7 points. In total 30 points.

Give careful and detailed answers to the questions!

ANSWER QUESTIONS 1 AND 2

---

1. An algorithm has the running time defined by the recurrence

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ 2T(n/2) + \Theta(n) & \text{if } n > 1 \end{cases}.$$

   Solve the recurrence. Be as exact about your derivation as you can. Which algorithms do you know that have this recurrence?

2. Explain the amortized analysis techniques

   (a) aggregate analysis,

   (b) accounting method, and

   (c) potential method.

   Illustrate each of these approaches using the extended set of stack operations (PUSH(S, k), POP(S), and MULTIPOP(S, k)) as an example.

ANSWER TWO OF QUESTIONS 3–6

3. What does open adressing mean in hashing? Explain the following techniques for computing the probe sequences in open addressing:

   (a) linear probing

   (b) quadratic probing, and

   (c) double hashing.

   How many distinct probe sequences are there for each of these methods? Which advantages and drawbacks do the techniques have?

4. Consider tossing identical balls randomly into b bins. Tosses are independent and on each toss the ball is equally likely to end up in any bin. The probability that a tossed ball lands in any given bin is $1/b$.

   How many balls must we toss until every bin contains at least one ball?

5. Computing optimal binary search trees using dynamic programming.

6. Huffman codes for data compression.