Aalto University, Department of Computer Science
Pekka Orponen

T-79.4202 Principles of Algorithmic Techniques (5 cr)
Exam Mon 14 Dec 2015, 1–4 p.m.

Write down on each answer sheet:
- Your name, degree programme, and student number
- The text: "T-79.4202 Principles of Algorithmic Techniques 14.12.2015"
- The total number of answer sheets you are submitting for grading

Note: You can write down your answers in either Finnish, Swedish, or English.

1. In the *Towers of Hanoi* puzzle one is given a board with three pegs A, B, C, and n wooden disks of different diameters that are initially stacked on peg A in decreasing order of diameter (see Figure 1). The task is to move the disks, one at a time, to the corresponding configuration on disk C, so that at no point in the move sequence is a wider disk placed on top of a narrower one. Peg B may of course be used as an auxiliary "intermediate landing site".

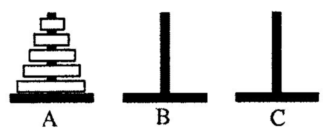

Figure 1: Initial configuration of Towers of Hanoi puzzle with 5 disks.

   (a) Assume you are given an elementary move operation $m(X,Y)$ that moves the topmost disk from peg X to peg Y. Using this operation as a subroutine, design a divide-and-conquer algorithm $M(n;X,T,Y)$ that moves (legally!) the n topmost disks from peg X to peg Y using peg T as an auxiliary site. (*Hint:* Consider first the $n-1$ topmost disks on peg X.) In particular, then, a call $M(n;A,B,C)$ solves the Towers of Hanoi puzzle for n disks. For instance, in the case of two disks, the call $M(2;A,B,C)$ should produce the sequence of elementary moves "$m(A,B);m(A,C);m(B,C)$".

   (b) Design a recurrence equation that describes the number $T(n)$ of elementary single-disk moves generated by the call $M(n;A,B,C)$, as a function of n. Solve this recurrence to get a closed-form expression for $T(n)$. *12p*

2. (a) Compute $d = \gcd(1917, 2016)$ and find integers x, y such that $1917x + 2016y = d$.

   (b) Find the inverses (if they exist) of: 7 mod 13, 14 mod 41, 7 mod 23, 14 mod 49. *12p*

3. Consider the following two-player game: a sequence of n coins, with positive integer values $v_1, \dots, v_n$ is laid on a table, and the players take turns in picking up either the leftmost or the rightmost remaining coin in the sequence. The game ends when all the coins have been picked up, and the player whose collected coins have a bigger total value, wins. Design an $O(n^2)$ algorithm to determine, for a given sequence of coins, what is the maximum total value that can be achieved by the player who takes the first turn. (*Hint:* Compute systematically the maximum total value achievable by the player in turn, when only a given middle interval of the coins remains.) *15p*

4. A directed acyclic graph G is *confluent*, if any two vertices v_1, v_2 in G which have a common ancestor u also have a common successor w. (I.e. if there are paths from some u to both v_1 and v_2, then there are also paths from v_1 and v_2 to some w. Vertices u, v_1, v_2 and w do not need to be distinct, thus e.g. a single "line" of vertices is trivially confluent.) Design a linear-time algorithm that determines whether a given DAG G is confluent. Justify the correctness and complexity of your algorithm. (*Hint:* Draw some examples of confluent and non-confluent DAGs. Characterise the confluence condition in terms of the leaves of G, i.e. the vertices with no outgoing edges.) *15p*