MALMÖ UNIVERSITY
Computer Science

Advanced Programming
Fall 2012

Advanced Programming Exam

2 November 2012 kl. 8.15–13.15

Instructions: Treat at most one question per sheet, write only on one page of each sheet and mark every page with your initials. Do not use red ink! Write legibly and comment your work extensively, this can give you extra marks even if the result is wrong. The exam contains 5 question and each question is worth 10 points.

Allowed Material: The course books (Dasgupta, Papadimitriou, Vazirani. *Algorithms*; Skiena, Revilla. *Programming Challenges*; Bentley. *Programming Pearls*), course notes and other printed material, pen or pencil and paper. **Good Luck!**

---

1. Let a and b be two integer numbers, a represented with n bits and b represented with m bits. Assume that two integers can be multiplied in constant $O(1)$ time, give an $O(m)$ time algorithm to compute a^b. (10p)

   (Hint: Use divide and conquer, dividing b into $b = 2 \cdot (b /\!\!/ 2) + 1$, if b is odd, and $b = 2 \cdot (b /\!\!/ 2)$, if b is even. The operation $/\!\!/$ represents integer division.)

2. The following DNA-sequence

   `CCAAGTTAAAACCAAGTTAAAACCACGTTCAAAGGGTTTTTAAACCAAAGTTAAACCAAGTTGTTAACCGTTA`

   is to be encoded in binary, i.e., using the letters 0 and 1. It consists of 30 `A`'s, 14 `C`'s, 10 `G`'s and 19 `T`'s.

   (a) What is the shortest binary encoding of this string and how many binary digits does it consist of? (6p)

   (b) Give binary codes for each letter `A`, `C`, `G` and `T`. (4p)

3. Consider a game between two players A and B. To win the game A or B has to win k points. Denote by $P(i, j)$ the probability that A will win if he/she needs another i points to win and B needs another j points. P is defined by

$$P(i,j) = \begin{cases} 0 & \text{if } i > 0 \text{ and } j = 0. \text{ B has won} \\ 1 & \text{if } i = 0 \text{ and } j > 0. \text{ A has won} \\ \frac{P(i-1,j)+P(i,j-1)}{2} & \text{if } i > 0 \text{ and } j > 0. \end{cases}$$

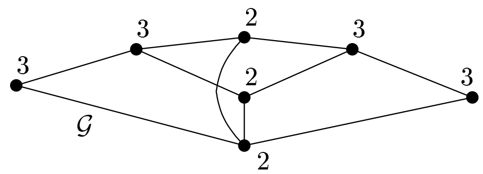
   (a) Construct an algorithm that computes $P(n, m)$ with the use of dynamic programming. (5p)

   (b) Analyze the complexity of your algorithm in terms of n and m. (2p)

   (c) Use your algorithm to compute $P(2, 3)$. (3p)

Please turn page!

4. Let $\mathcal{G}$ be an unweighted graph. Label each vertex v with the distance from v to its farthest node. We define the *kernel* of a graph to be the vertices with the smallest label. Here is a small example:



The kernel contains the vertices labelled 2 in the figure.

Develop an efficient algorithm that computes the kernel of an unweighted graph $\mathcal{G}$. Your algorithm should require at most $O(n^3)$ time. (10p)

5. Given n points in the plane, there exist algorithms that compute the closest pair among the n points in $O(n \log n)$ time.

Develop an algorithm also running in $O(n \log n)$ time that finds the *second closest pair* among n points in the plane. (10p)

(Hint: You may run a closest pair algorithm several times if you like.)