582630 Design and Analysis of Algorithms (4 cu) Fall 2012
Instructor: Dr. Mikko Koivisto, University of Helsinki
Exam 18.10.2012

Please solve all the six problems given below. **Justify your solutions by proper argumentation.** Show your work—partial credits will be given. Be clear and neat. Use either English, Swedish, or Finnish. You have two and a half hours in total—plan your time usage! Include your name and identity number, and the name of the course in every paper sheet you return. Number the problems clearly.

1. (max 9 points) Give a function $f(n)$ such that $T(n) = \Theta(f(n))$,

(a) when $T(1) = 1$ and $T(n) = 3\,T(n/3) + n$ for $n > 1$,
(b) when $T(1) = 2$ and $T(n) = 2\,T(n/3) + \log_2 n$ for $n > 1$.

You may confine your analysis to exact powers of 3.

2. (max 9 points) The *segmentation problem* is as follows. For each pair (i, j) of integers, $1 \le i \le j \le n$, we are given a cost $c(i, j) > 0$, and the task is to find an increasing sequence of cutpoints $i_1, i_2, \ldots, i_k \in \{1, 2, \ldots, n-1\}$ so as to minimize the total cost $\sum_{s=0}^{k} c(i_s + 1, i_{s+1})$, where $i_0 = 0$ and $i_{k+1} = n$. Give an algorithm that solves the problem in $O(n^2)$ time.

3. (max 9 points) The *set-partition problem* takes as input a set S of numbers. The question is whether the numbers can be partitioned into two sets A and $S \setminus A$ such that $\sum_{x \in A} x = \sum_{x \in S \setminus A} x$. Show that the set-partition problem is NP-complete. (*Hint:* Reduce from SUBSET-SUM that you may assume is known to be NP-complete).

4. (max 9 points) Show that the following algorithm is correct and runs in $O(n^2)$ time:

```
Input: An undirected graph G in adjacency list representation,
       vertices labeled by 1, 2, ..., n.
Output: "YES" if there is a simple cycle of four edges in G,
       and "NO" otherwise.
Initialize an array C[1..n, 1..n] to all-zeros.
For i = 1 to n
  For each pair (j, k) of distinct neighbors of i in G
        if C[j, k] == 1
          Return "YES"
        else
          C[j, k] = 1
Return "NO"
```

5. (max 9 points) Show that there is a polynomial-time randomized algorithm that given a 3-CNF-SAT formula ϕ as input, outputs a truth assignment that, in expectation, satisfies at least 7/8 of the clauses of ϕ. You may assume that each clause in the formula consists of exactly three distinct literals. You may also assume the availability of a routine RANDOM(0, 1) that returns 0 with probability 1/2 and 1 with probability 1/2. (*Hint:* Consider a truth assignment drawn uniformly at random.)

6. (max 9 points) Suppose somebody gives you an algorithm $\mathcal{A}$ that, given an n-vertex undirected graph and one of its vertex s as input, *decides* in $O(\beta^n)$ time whether the graph contains a hamiltonian path that starts from s. Here $\beta > 1$ is a constant independent of n. Using $\mathcal{A}$ as subroutine, give another algorithm that, given the graph and the vertex s as input, *constructs* such a hamiltonian path when one exists. The algorithm should be as fast as possible. (Time bound $O(\beta^n n^k)$, with some constant k, yields 5 points; $O(\beta^n n)$ yields 7 points; $O(\beta^n \log n)$ yields full points.)