Final Exam — Introduction to Algorithms (CS 300)

June 21, 2013, 10:00 to 12:00

- Before you start: Write your name and student number on *every answer sheet.*
- This is a *closed book* exam. You are not allowed to consult any book or notes.
- To ensure a quiet exam environment, we will not answer questions during the exam. If you think there is a mistake in the question, explain so on your answer sheet, and use common sense to answer the question.
- The questions have to be answered in *English.* Write clearly!

Problem 1: (30 pts) The following table shows the relative frequency of the 26 letters in the German language:

E	17.4 %	N	9.8 %	I	7.5 %	S	7.3 %	R	7.0 %	A	6.5 %
T	6.2 %	D	5.0 %	H	4.8 %	U	4.4 %	L	3.4 %	C	3.1 %
G	3.0 %	M	2.5 %	O	2.5 %	B	1.9 %	W	1.9 %	F	1.7 %
K	1.2 %	Z	1.1 %	P	0.8 %	V	0.7 %	J	0.3 %	Y	0.04 %
X	0.03 %	Q	0.02 %								

Construct an optimal prefix code for the 26 letters. Your answer should only contain the code, represented in a reasonable way.

Problem 2: (30 pts) A palindrome is a string that reads the same forwards and backwards, like `X`, `373`, `noon`, `redivider`, or `amanaplanacatahamayakayamahatacanalpanama`. Any string can be written as a sequence of palindromes. For example, the string `bubbaseesabanana` ('Bubba sees a banana.') can be decomposed in several ways; for example:

- `bub + baseesab + anana`,
- `b + u + bb + a + sees + aba + nan + a`,
- `b + u + bb + a + sees + a + b + anana`,
- `b+u+b+b+a+s+e+e+s+a+b+a+n+a+n+a`.

Describe an efficient algorithm to find the minimum number of palindromes that make up a given input string. For example, given the input string `bubbaseesabanana`, your algorithm would return the number 3.

You will receive 8 points if you write "I don't know" and nothing else.

Problem 3: **(a)** (5pts) Consider two decision problems ProblemA and ProblemB. You want to prove the reduction

$$\textsc{ProblemA} \longrightarrow \textsc{ProblemB}.$$

To prove this, you have to give an algorithm. For which problem do you need to give an algorithm, and what are the requirements for this algorithm?

(b) (5 pts) Consider the following problem DoubleHamiltonianCycle: Given an undirected graph G, does G contain a closed walk that visits every vertex in G exactly *two times*? (It is allowed to use an edge more than once.)

A friend has proven two reductions involving this problem:

$$\textsc{ProblemA} \longrightarrow \textsc{DoubleHamiltonianCycle},$$
$$\textsc{DoubleHamiltonianCycle} \longrightarrow \textsc{ProblemB}.$$

Which of these two reductions could be used to prove that DoubleHamiltonianCycle is NP-hard? What would be the requirement for ProblemA or ProblemB?

(c) (10 pts) Explain why DoubleHamiltonianCycle is in NP.

(d) (20 pts) Prove that DoubleHamiltonianCycle is NP-hard.