582487 Data Compression Techniques (Spring 2012)
Exercises 2 (1 February)

Solve the following problems before the exercise session and be prepared to present your solutions at the session.

1. Implement the Shannon–Fano algorithm (Exercise 1.3) to construct a prefix code for the set $\mathbb{N}$ of non-negative integers with the following probability distributions:

   (a) $P(n) = (1 - p) \cdot p^n$, where $p = 2^{-1/4} \approx 0.841$.

   (b) $P(n) = (n + 1)^{-2}/\zeta(2)$, where $\zeta(2) = \pi^2/6 \approx 1.645$ (ζ is the Riemann zeta function).

   Give the codes for the numbers 0–9.

2. Assume that the model of computation supports the following operation in constant time:

   - Read up to w consecutive bits starting from any position in a binary string, and interpret the bits as a binary representation of an integer.

   Suppose we have encoded integers in the range $[0..2^w)$ using the γ-code. Show how an integer n can be decoded in $\mathcal{O}(\log \log n)$ time.

   Hint: Use the fact that γ-code is canonical (see Problem 1.5).

3. What is

   (a) the entropy

   (b) the average Huffman code length

   of the following distribution

symbol	a	b	c	d	e	f	g
prob.	1/6	1/5	1/20	1/5	1/120	1/8	1/4

4. On slide 48 of the lecture notes, there is a claim that the rounding scheme cannot cause undesirable overlaps between intervals. Explain why this is true.

5. A detailed example of the encoding process for arithmetic coding is described on slides 45–52 of the lecture notes, but the decoding process is only outlined on slide 53. Describe the decoding process for that example using a similar level of detail as in the encoding process.

6. To decode the code string 101110 produced in the example on slides 45–52 in the lecture notes, one has to know the length of the source string. Slide 43 describes two ways to produce a code string that can be decoded without knowing the length. Apply both of them to the example on slides 45–52.