Heads and Tails, summer 2009

PROF. DR. JOACHIM VON ZUR GATHEN, DANIEL LOEBENBERGER

5. Exercise sheet
Hand in solutions until Sunday, 24 May 2009, 24:00h.

Exercise 5.1 (Trits et al.). (2 points)

In the course you learned how to compute the entropy of a random trit. Here we will generalize the problem slightly: Suppose you wish to generate a random n-it, where $n \in \mathbb{N}_{\geq 2}$. That is you wish to generate a number randomly from the set $\{1, \ldots, n\}$ using random bits. Compute the entropy of a n-it. $\boxed{2}$

Exercise 5.2 (Entropy and Huffman trees). (6 points)

We are given an alphabet $\mathbb{A} = \{A, B, C, D, E, F\}$ with the following frequency distribution:

Letter	A	B	C	D	E	F
Frequency	5	18	10	15	45	7

(i) Compute the corresponding entropy. $\boxed{1}$

(ii) Using the same number of bits to encode each letter, how many do we need? $\boxed{1}$

(iii) What is the expected length of an n-letter message with this encoding? $\boxed{1}$

(iv) How can you do better? $\boxed{3}$

Exercise 5.3 (Entropy of the English language). (13 points)

The following table gives the frequency distribution of the letters in English.

Letter	A	B	C	D	E	F	G	H	I
Frequency	8.04	1.54	3.06	3.99	12.51	2.30	1.96	5.49	7.26

Letter	J	K	L	M	N	O	P	Q	R
Frequency	0.16	0.67	4.14	2.53	7.09	7.60	2.00	0.11	6.12

Letter	S	T	U	V	W	X	Y	Z
Frequency	6.54	9.25	2.71	0.99	1.92	0.19	1.73	0.09

(i) What is the entropy of English? $\boxed{1}$

(ii) What is the maximal entropy for a 26-letter alphabet? $\boxed{1}$

$\boxed{1}$ (iii) Compute the *redundancy* of English, *i.e.* the entropy distance between English and a uniformly-distributed 26-letter language.

$\boxed{4}$ (iv) Give a Huffman encoding of English according to this frequency distribution.

$\boxed{2}$ (v) Have fun with the Java applet available at this URL:
`http://math.ucsd.edu/~crypto/java/ENTROPY/`
What entropy do you obtain?

$\boxed{2}$ (vi) Why is it lower than the previously computed entropy?

$\boxed{2}$ (vii) Does it mean that we can compress an English text in approximately 1.2 bits per letter? Why aren't such extreme compression techniques not used in practice?

Exercise 5.4 (Linear congruential generators). (7 points)

We consider the linear congruential generators with $x_i = (ax_{i-1} + b) \operatorname{rem} m$.

(i) Compute the pseudorandom sequence of numbers resulting from

$\boxed{2}$ (a) $m = 10$, $a = 3$, $b = 2$, $x_0 = 1$ and

$\boxed{2}$ (b) $m = 10$, $a = 8$, $b = 7$, $x_0 = 1$.

What do you observe?

$\boxed{3}$ (ii) You observe the sequence of numbers

$$13, 223, 793, 483, 213, 623, 593, \ldots$$

generated by a linear congruential generator. Find matching values of m, a and b.