Chair for Algorithms
and Data Structures
Prof. Dr. Hannah Bast
Marjan Celikik

Search Engines WS 09/10

http://ad.informatik.uni-freiburg.de/teaching

UNI FREIBURG

Exercise Sheet 4

complete until Tuesday, November 17th

Exercise 1

Write the following program. Generate a random permutation of the integers $1, \ldots, n$ with a single cycle and store it in an array A. Then sum over all elements of the array in two ways. (1) With a simple scan over the array, from the first to the last position. (2) By following the permutation, that is, start at A[1], then go to A[A[1]], etc. Check that (1) + (2) do the same thing, and compare their running times for $n = 10^i$, where $i = 1, \ldots, 8$. Explain the differences in the running times you observe (also quantitatively).

Exercise 2

Prove that the Elias code is prefix-free. Then prove that any iteration of the Elias scheme (Elias-Gamme code, etc.) is also prefix free.

Discuss how the best choice for k in Elias encoding depends on the given data. In particular, give an example of data for which $k = 1$ is the best choice, and give an example of data where $k = 2$ is the best choice.

Exercise 3

Prove that the entropy of a discrete random variable is maximal when all values are equally likely. After the proof, give an intuitive explanation of why this makes sense.

Exercise 4

Write the following program. Generate a random inverted list of size m, with doc ids from a range $[1..n]$. Then write the list to a file in two ways. (1) Uncompressed, as a sequence of 4-byte ints. (2) Compressed with a non-trivial encoding scheme of your choice (but don't just call a library, you have to implement the encoding yourself). Then compare the running times for reading the uncompresed list from (1) and the compressed list from (2). Also time how long it takes to decompress the list. Important note: Whenver you measure running times for reading data from disk, you have to clear the disk cache before, as discussed in the lecture.

Exercise 5* (Optional / Bonus)

Prove that there is no prefix-free encoding scheme with a code length of at most $\log_2 x$ bits for number x, for all positive integers x. (Note that in the lecture we had a few coding schemes that achieved $C \cdot \log_2 x$ bits, for some $C > 1$.)