Text Algorithms MTAT.03.190 – Home Exam

Time period: Friday, January 4 – Monday, to Monday, Jan 14th 20:00.

Exam is personal. No collaboration with co-students or anyone else is allowed. All the material – from web, books, libraries, code - are ok to be used. Your textual descriptions (submission of the exam) have to be explicitly written by you in your own words. In case of any copy-paste, citation, or other material use, the source has to be provided. If you use software/sources from net, you must explicitly say so in the text.

Front page must have on the top right corner – your name and email. Every page must likewise have your name and page nr (e.g 3/5 – page 3 out of 5) .

Submission: You must submit the exam as a Word, PDF or tar.gz/tgz , zip file through Homework submission form by 20:00 on Monday, January 14th. **Every delayed hour will subtract 1 point from the grade.**

You must collect as a minimum 50% of points from the exam.

1. (15p) **Describe** the bit-wise parallel matching of regular expressions with errors (indels, substitutions). Write a maximum 2 page "extended" abstract about this topic.

2. (15p) **Text indexing using BWT transform**

    1. Decode the following Burrows-Wheeler encoded sentence (a question). **yeseessy_rrrhhhlnittw_d__nnoguheeeiisee__si_?tt**
    2. Describe the decoding process.
    3. Describe a BWT based text index that allows to simulate suffix trees over the BWT encoded text. I.e. – how the BWT encoded text allows to make a compressed text index.

    Be concise – brief and strict. Also pay attention to the illustration of the main concepts. Try to make at least one illustration/figure to describe this index.

3. (10p) Practical assignment. Download the Exam_texts_corpus.tgz linked from course homepage. http://courses.cs.ut.ee/2010/text/uploads/Main/Exam_texts_corpus.tgz These are three books of jokes from Project Gutenberg.

    a. Extract individual jokes, each into a separate file numbered 001.txt .. nnn.txt . Use regular expressions (perl, python, grep, awk, etc tools ...) for text parsing, provide the parser. (this does not have to be perfect, satisfy with respectable quality). Describe how many jokes. (i.e. extract all intro's, etc irrelevant parts)

    b. Count word frequencies, exatract lists of words sorted by frequency (all jokes; each book separately) and words "most specific" to each book. Provide top-25

lists. If you have a general text – pick some book or set of books from Gutenberg – then you can also identify jokes specific word list.

c. **Propose a way to group jokes by topic or content similarity**. E.g. search for a list of words and then group by ocurrences of the same (rare) words or semantically related topcis. **Pick an interesting joke and fetch most similar jokes.**

Happy Exam!