DM823 String Algorithms

Fall 2010

Department of Mathematics and Computer Science
University of Southern Denmark

October 7, 2010

Project Addendum

This document is an addendum to the project text, and gives the details of the experiments which should be carried out.

Data Sets

A collection of 16 texts of various types and alphabet sizes has been created, and can be found in the file `ProjectDataSets.zip`. Each text is a single file, and the contents of the files are described in the file `INDEX.TXT`.

Experiments

Let x ($1 \leq x \leq 4$) be the number of members of the group. You are to perform the following for each of at least $6 + 2x$ of the texts (of your own choice).

For each text, the pattern to search for should be m characters starting at a random position in latter half of text. All occurrences should be found (not just the first). The values of m used should be: 4, 8, 16, 128 and $n/8$, where n is the text length.

For each combination of

- Algorithm implemented
- Pattern length m
- Text chosen

do the following:

1. Load the text into an array of bytes.

2. Select the pattern in the text (remember to align to a multiple of the bytes-per-character value) and copy it from the text into another array.
3. Read the time.
4. Preprocess the pattern (if the algorithm uses preprocessing).
5. Do the search for all occurrences.
6. Read time again and calculate elapsed time.

Repeat from point two enough times for the sum of the elapsed times to be around five seconds or more. However, at least five repetitions should always be done. Then take the average elapsed time (i.e., sum of elapsed times divided by number of repetitions).

During experiments, you should just count the number of occurrences (and report a single number in the end), in order not to spend time on output. During development, you for debugging reasons may also want to report the position of occurences (and test against the brute force method). Compile with maximal optimization flags (if available—Java has none).

Presentation of Results

You should display your results using grouped/clustered bar charts (see e.g. example two in example section of `http://www.burningcutlery.com/derek/bargraph/` for a definition). You should make one chart per text. The y-axis should be running time (average over all repetitions of search, cf. above), and the x-axis should contain groups of experiments, with the pattern length being the same within a group, and the members of a group being the different algorithms implemented.

To get a reasonable resolution in the diagrams: if the brute force is more than three times slower than the second slowest method, leave it out of the chart (but report in writing the factor it is slower than the second slowest).