Project in String Processing Algorithms

Spring 2011, period III

Juha Kärkkäinen

Who is this course for?

- Master's level course in Computer Science, 2 cr
- Continuation of String Processing Algorithms course
- Requires some programming experience
- Subprogram of Algorithms and Machine Learning
  - Together with String Processing Algorithms one of the three special course combinations, one of which must be included in the Master's degree.
- Suitable addition to Master's degree program for Bioinformatics, particularly for those interested in biological sequence analysis
- Good fit for Subrogramme of Software systems

Course structure

- Three main tasks
  - Implementation of string processing algorithms
  - Experimental analysis and/or comparison of the algorithms
  - Presentation of the results as a poster
- Each task has about the same weight in grading
- Can be done in groups of at most four
  - Each group member implements different algorithms

Algorithm implementation

Each student implements one or more algorithms or variations

- The number depends on the degree of difficulty
- Can be based on existing implementations
- Any programming language, provided that:
  - Compiles and runs on department computers
  - Same within a group
- Important qualities:
  - correct, well tested
  - readable, well documented
  - efficient, well tuned

Algorithm implementation (continued)

Return to instructor:

- Implementation code
- Scripts for compiling and running tests
- Documentation
  - description of what was done: existing code used, main design decisions, tuning details etc.
  - guidance for understanding the code
  - instructions for compiling and running
  - format is free, even comments to code is OK
- By email in a single package (zip, tar.gz, or something like that)

Experiments

- The purpose of the experiments:
  - Determine the performance of algorithms under different conditions
  - Find best algorithms, variations or parameter settings
- Choice of test data is important
  - Try to find best and worst cases for each algorithm.
  - Compare theory and practice.
  - Use generated, artificial data for fine control of parameters, real world data for real world performance.
  - Avoid too trivial experiments. For example, exact string matching time is trivially linear in the length of the text.
- Mainly joint responsibility of a group, but each student should make sure that her or his algorithms are well represented.

Poster

- Includes:
  - Description of the problem
  - Description of algorithms and implementations
  - Experimental setting (repeatability)
  - Experimental results and their interpretation
- Presented to an audience of other students and staff of the department
  - Not all have taken the String Processing Algorithms course (recently)
- Visual clarity is important
  - Avoid too much detail, include only main points and results. Additional details may be explained verbally.
  - Use figures, graphs, colors, etc.
- See examples

Topic: Exact String Matching

- KMP, Shift-Or, Horspool, BNDM, BOM, ...
- ESMAJ: `http://www-igm.univ-mlv.fr/~lecroq/string/`
- B. Ďuriana, J. Holub, H. Peltola, and J. Tarhio: Improving practical exact string matching. Information Processing Letters (IPL), 110(4): 148–152, 2010. `http://dx.doi.org/10.1016/j.ipl.2009.11.010`

Topic: Multiple Exact String Matching

- Aho-Corasick
- Multi-pattern versions of Shift-Or, Horspool, BOM, Karp-Rabin, ...
- L. Salmela, J. Tarhio, and J. Kytöjoki: Multipattern string matching with q-grams. Journal of Experimental Algorithmics 11, Article 1.1 (February 2007). `http://doi.acm.org/10.1145/1187436.1187438`

Topic: Approximate String Matching

- Standard dynamic programming, Ukkonen's cut-off heuristic, Myers' bitparallel algorithm, filtering algorithms, ...
- G. Navarro: A guided tour to approximate string matching. ACM Computing Surveys 33(1): 31–88, 2003. `http://doi.acm.org/10.1145/375360.375365`
- L. Salmela and J. Tarhio: Approximate String Matching with Reduced Alphabet. Workshop on Algorithms and Applications, LNCS 6060, Springer 2010. `http://dx.doi.org/10.1007/978-3-642-12476-1_15`

Topic: String sorting

- String quicksort, string mergesort, MSD radix sort, ...
- R. Sinha and A. Wirth: Engineering burstsort: Toward fast in-place string sorting. Journal of Experimental Algorithmics 15, Article 2.5 (March 2010). `http://doi.acm.org/10.1145/1671973.1671978`
- Cache misses are important

Other topics

- string search trees
- suffix array construction
- ...

Tentative schedule

- 18.1. Formation of groups, selection of topics.
- 25.1. Finalization of algorithms and variations.
- 1.2. Additional details on implementations.
- 8.2. Initial design of experiments.
- 15.2. Implementations (nearly) finished, final design of experiments, initial design of poster.
- ??.2. Return of implementations.
- 22.2. Final design of poster.
- ??.2. Poster presentation.