Project in String Processing Algorithms
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 three person
  - Role of each person in the group need to be reported when returning the code
Algorithm implementation

• Each student should implement some part of the core algorithms
  ○ 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
Return to instructor:
- Implementation code
- Scripts for compiling and running tests
- Documentation
  - description of what was done: existing code used, main design
  - the role of each person in the group
  - 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.
- Mainly joint responsibility of a group, but each student should make sure that her or his algorithms are well represented.
• Includes:
  o Description of the problem
  o Description of algorithms and implementations
  o Experimental setting (repeatability)
  o Experimental results and their interpretation

• Presented to an audience of other students and staff of the department
  o Not all have taken the String Processing Algorithms course (recently)

• Visual clarity is important
  o Avoid too much detail, include only main points and results.
  o Additional details may be explained verbally.
  o Use figures, graphs, colors, etc.

• See examples
Tentative schedule

- 17.1 Formation of groups, selection of topics.
- 24.1 Main structure of the algorithms studied, work plan ready
- 31.1 Algorithms studied in the implementation level detail
- 7.2 Initial design of experiments
- 14.2 Implementations nearly finished, final design of experiments, initial design of poster
- 17.2 Return of implementations (noon)
- 21.2 Final design of poster
- ???.2 Poster presentation
Topics

• Previous year’s topics here: www.cs.helsinki.fi/juha.karkkainen/opetus/10s/spa/opening.pdf