CSE|BMI 577 – Processing of Strings and Sequences

Last updated: 2019-01-18 11:46

Course Information

- Date(s)/Time(s): MoWeFr 3pm-3:50pm
- Duration: 14 weeks (42 lectures)
- Location: Norton 213
- Credits: 3
- Instructor: Dr. Jaroslaw Zola, Assistant Professor
  Department of Computer Science and Engineering
  Department of Biomedical Informatics
  Email: jzola@buffalo.edu
  Web: http://www.jzola.org/

CSE Focus Area: Software and Information Systems (SW)

Course Description

This course is intended for students interested in learning efficient techniques for processing and analyzing large text collections, such as large-scale system logs, massive text corpora, medical records, or databases of DNA and protein sequences. The main focus is on fast algorithms and data structures for strings and sequences, including pattern matching, pairwise comparison, indexing and searching, as well as probabilistic methods, like fingerprinting and hashing. The theoretical component is complemented by practical considerations regarding efficient implementations of the discussed algorithms, and their applications in the real-world systems. The example applications include tools like UNIX grep, frameworks for plagiarism detection, as well as tools driving computational biology (e.g., BLAST, read mappers, DNA assemblers, etc.). The course has also a programming component, in which students implement (in their language of choice C/C++, Python or Java) small but fully functional text processing applications.

This course is the Software and Information Systems **focus area** course at CSE.

Course Organization

The course consists of a series of lectures covering multiple algorithms on strings and sequences, including their design, analysis and real-world applications. Lectures are complemented with a programming assignments exposing practical aspects of the covered material. The course outline is provided below:

1. Basic notation and techniques in reasoning about string algorithms (1wk)
2. Exact pattern matching: Knuth-Morris-Pratt, Boyer-Moore and Aho-Corasick algorithms (1wk)
3. Inexact matching and pairwise sequence comparison: Smith-Waterman and Needleman-Wunsch algorithms, spliced and syntenic alignment (2wks)
4. Winnowing and fingerprinting for documents comparison (1wk)
5. Suffix Trees: construction, querying, applications (2wks)
6. Suffix Arrays and LCP arrays: construction, querying, applications (2wks)
7. BWT and FM-Index: construction, querying, applications (2wks)
8. Locality sensitive hashing for text processing (1wk)
9. Alignment-free sequence comparison (1wk)
10. Text processing applications: plagiarism detection, DNA clustering and assembly (1wk)

Table 1: Points to grade mapping.

Percentage score	Grade	Quality points
90-100	A	4.0
85-89	A-	3.67
80-84	B+	3.33
75-79	B	3.0
70-74	B-	2.67
65-69	C+	2.33
60-64	C	2.0
55-59	C-	1.67
50-54	D	1.0
0-49	F	0.0

Course Prerequisites

The course has no specific prerequisites for CSE graduate students. For BMI students, "BMI503: Biomedical Informatics Systems, Databases and Software Methods" is the main prerequisite. The course requires some basic experience in synthesis and analysis of algorithms, at least at the level of "CSE250: Data Structures and Their Algorithms." The course has a programming component, hence a rudimentary ability to learn new programming constructs is expected. While the course is programming language oblivious, to simplify discussions and grading we will be using C++14, Python or Java.

Program Outcomes

Upon completion of this course you will:

- Gain basic skills required to design and analyze algorithms on strings and sequences.
- Be able to select and apply proper algorithms to process large text corpora, including large system logs, text documents, biomedical records, and DNA/protein databases.
- Be able to implement string processing algorithms at various levels of complexity (e.g. with and without indexing).

Course Requirements

The course has three requirements:

1. Midterm exam testing your understanding of the most basic string algorithms and the ability to reason about their performance and applicability.
2. Programming assignments exposing you to the practical aspects of the covered material. Each assignment will be a mini-project implementing a small text processing application (e.g. grep, etc.).
3. Final exam testing your overall understanding of the material.

Grading Policy

The final grade will be weighted average: 20% midterm exam, 30% final exam, 50% programming assignments. The number-to-letter grade mapping will be done as indicated in Table 1 below.

Incomplete Grades

In general, no incomplete grades ("IU") will be given. However, in special circumstances that are truly beyond your control and justify incomplete grade, we will follow the university policy on incomplete grades, available at: http://grad.buffalo.edu/study/progress/policylibrary.html .

Course Materials

This course does not have a required textbook. However, the following books are highly recommended, as the course is roughly based on their content:

1. D. Gusfield, “Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology,” Cambridge University Press, 1997.
2. M. Crochemore, W. Rytter, “Jewels of Stringology,” World Scientific Publishing Company, 2002.

Additional readings (e.g. papers, tutorials, etc.) will be referenced thought out the course as needed.

Additional Course Resources

For the duration of the course you will have access to multiple resources to help you with assignments and preparation to exams. This includes a web page where lecture slides and videos will be posted, Piazza discussion forum, assignments evaluation system (autograder), and a virtual image or container with Ubuntu Linux and basic development environment. Details will be provided during the first lecture.

Academic Integrity

You **must** familiarize yourself with the university policies on academic integrity available, at http://grad.buffalo.edu/study/progress/policylibrary.html . If you are a CSE student, you must also be familiar with the CSE departmental policies available from here . **We take these policies very seriously!**

Any violation of these policies, including but not limited to cheating on any course deliverable (e.g. homework project, exam, etc.), will result in **automatic failure of the course**. There will be no leniency! This is not because I am mean or do not like you. This is because I value and protect hard working students who do not cheat.

If you decide to use a code or other result from some external source, e.g. an open source project, you must include a proper and clearly visible attribution in your product (you are encouraged to contact your instructor to check if the artifact you plan to use is admissible).

Accessibility Resources

If you have any disability that requires reasonable accommodations to enable you to participate in this course, please contact the Office of Accessibility Resources, 60 Capen Hall, Phone: (716) 645-2608, and also the instructor. The office will provide you with information and review appropriate arrangements for reasonable accommodations.