02-[46]14: String Algorithms

Carl Kingsford
Spring 2020

1. Course Information

1.1 Key details:

Instructor: Carl Kingsford, *Office:* GHC 7719, *Phone:* 412-268-1769, *Email:* carlk@cs.cmu.edu

Office hours: Will be posted on the class webpage. You are encouraged to make use of office hours, and the TA will be as helpful as possible to explain material and homework questions and solutions.

Final: According to the university's schedule.

1.2 Course description

Provides an in-depth look at modern algorithms used to process string data, particularly those relevant to genomics. The course will cover the design and analysis of efficient algorithms for processing enormous collections of strings. Topics will include string search; inexact matching; string compression; string data structures such as suffix trees, suffix arrays, and searchable compressed indices; and the Borrows-Wheeler transform. Applications of these techniques in genomics will be presented, including genome assembly, transcript assembly, whole-genome alignment, gene expression quantification, read mapping, and search of large sequence databases. No knowledge of biology is assumed; programming proficiency is required.

1.3 Course objectives

The course will focus on describing algorithms that work with strings and string-like data in a rigorous way. We will typically describe why the algorithms are correct and give proofs (sometimes abbreviated or sketched) for runtime. For each major topic, we will describe at least one application from genomics that motivates the developed algorithms. We will include examples from other application areas as well. We have the following objectives:

- Learn various algorithmic techniques and data structures for efficient processing of string data, including suffix trees, suffix arrays, Borrows-Wheeler transforms.
- Understand the why these algorithms and data structures work.
- Learn to apply and extend these algorithms and data structures.
- Learn about the practical application of these techniques, especially in genomics.
- At the end of this class, you should be familiar with much of the state-of-the-art in algorithms for strings, have familiarity with their use in practice, and have experience applying them to new problems.

1.4 Prerequisites

- Equivalent of 15-210 ("Parallel & Sequential Data Structures and Algorithms") or 15-351 ("Algorithms and Advanced Data Structures").

- Programming proficiency (the course will not include significant programming, but a few homeworks may include a programming component).

1.5 Textbooks

There is no required textbook because no one book covers all the topics we will discuss. However, the following books each provide good coverage of some of the topics:

- *Genome-Scale Algorithm Design: Biological Sequence Analysis in the Era of High-Throughput Sequencing* by Veli Mäkinen, Djamal Belazzougui, Fabio Cunial, Alexandru I. Tomescu
- *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology* by Gusfield
- *Algorithms on Strings* by Crochemore

1.6 Coursework

- (35%) several written or programming homework problem sets. We expect to have 4–6 written homework sets.
- (15% each) two midterms. The midterms will be in class on the dates announced on the first day of class. The second midterm will cover the material between the first midterm and the second midterm.
- (10%) participation and in-class quizzes. We may have short checkpoint quizzes in class to help gauge the class' understanding. Participation will be based on attendance and engagement in class.
- (25%) a final exam. The final will cover all the material from the course. It will be scheduled at the university-appointed time.

2. Tentative Topics

- **Exact string matching** (Gusfield, chapters 1-4): the Z-algorithm; Knuth-Morris-Pratt; Boyer-Moore; seminumerical string matching (Rabin-Karp).
- **Inexact matching** (Gusfield, chapters 10–13): Computing basic edit distance; string alignment in linear space; Four Russians speed up; Example application: BLAST.
- **String data structures:** Suffix trees/arrays (Gusfield, chapters 5–9): definition of suffix trees and arrays; Ukkonen's suffix-tree construction algorithm; applications of suffix trees and arrays. Compressed self-indices (i.e. data structures that support fast searching and reconstruction of the full sequence in sublinear space; these are the basis of current read-mapping techniques): Burrows-Wheeler transform; the FM-index. Minimizers, sparse suffix arrays, MEMs, SMEMs, MUMs, and whole genome alignment. Example application: MUMmer, Bowtie, BWA.
- **Multiple sequences:** motif finding, multiple pattern search, approximation algorithms for multiple sequence alignment (Gusfield, ch. 14). Example application: influenza phylogenies.
- **Compression, other compressed data structures:** String compression algorithms (e.g. Lempel-Ziv, Gusfield, chapter 7.17). RRR bit vectors, wavelet trees, Bidirectional BWT.

- **Hashing / randomization techniques for large string collections:** de Bruijn graphs and de Bruijn sequences, Locality sensitive hashing for strings, Nearest neighbor search with locality sensitive hashing, Random projection for motif finding, Bloom filters and Sequence Bloom Trees. Example application: Mash, mashmap, sequence Bloom trees.
- **State machines.** Regular expressions and FSAs, Context-free grammars, parsers, HMMs, Example application: gene finding.
- **String graphs.** Variant graphs, genome assembly, BWT for labeled trees and DAGs. Example application: read alignment to a population.
- **Other current research in "Big Genomics".** Topics TBD and vary semester-to-semester.

3. Policies

Homework policies:

- Homeworks are due at the start of class. **No late homework will be accepted** — turn in what you have completed.
- Answers to homework problems should be written concisely and clearly. You can lose points for both incorrectness and poor exposition. Homeworks must be typeset and submitted as PDFs. Instructions for submission will be posted on the course webpage.
- Homework problems that ask for an algorithm should present: a clear English description, an argument that the algorithm is correct, and an analysis of the running time. Please do **not** include complex pseudocode to explain your solution. Your goal is to explain the algorithm to a human, not a computer — as such detailed pseudocode or source code is usually *not* the best way to explain an algorithm. One way to think about how much detail to include is that you are trying to convince a skeptical reader that you know the correct solution. Another way to think about it: imagine you are a manager telling a programmer who works for you how to solve the problem; what would you tell them?
- If you use any reference or webpage or material from any other class, you must cite it, or we will consider this cheating. You are welcome to use general background and CS resources so long as you do not use material that gives the answer directly. You may lose points if your cited resources hew too closely to giving the answer to the problem.
- Regrade requests should be made **in writing** within 1 week of the regrade requests option on Gradescope being turned on. The entire homework or exam in question may be regraded, which may result in a higher or lower grade than originally returned.
- Depending on the lengths of the homeworks, we may employ a randomized grading strategy where we will grade only a random subset of the homework problems on any individual homework.
- You may discuss homework problems with classmates. You must list the names of the class members with whom you worked at the top of your homework. **You must write up your own solution independently!** "Independently" means — at least — that you cannot look at another person's homework, you cannot have them look at yours to see if it is correct, you cannot take detailed notes from a discussion and edit them into your homework, and you cannot sit in a group and continue discussing the homework while writing it up. The intent of this rule is: you can gather around a whiteboard with your fellow students and discuss how to solve the problems. Then you must all walk away and write the answers up separately. Note:

since the exams count much of your grade, there's little benefit in writing down a homework answer that you don't understand.

Unfortunately, each semester, we find some people who have copied each other's homeworks. Such instances are referred to the University according to the academic integrity violation policy.

- You may *never* use, look at, study, or copy any answers from previous semesters of this course.
- You must write all programming assignments on your own and cannot share code with other students or use code obtained from other students. In addition to manual inspection, we use an automatic system for detecting programming assignments that are significantly similar.

Exam policies.

- Students claiming an excused absence for an in-class exam or the final must supply documentation (such as a doctor's note) justifying the absence. Absences for religious observances must be submitted by email to the instructor during the first two weeks of the semester.
- Note it is SCS policy that in general job interviews are not a valid reason for missing an assignment or exam.
- Approved make-ups for midterms or the final that are not the result of illness or emergency must be scheduled a week before the exam.
- SCS asks us to "remind [you] NOT to schedule flights or plan to leave Pittsburgh at the end of the semester until [you] know [your] final exam schedule." The instructors have no control over when final exams are scheduled for any individual class.
- You may bring 1 page (double sided) of notes to each midterm and final. This page must be **hand written by you**. Use of others' "cheatsheets" is forbidden.

Classroom etiquette.

- Attendance in lecture is expected.
- To minimize disruptions and in consideration of your classmates, I ask that you please arrive on time and do not leave early. If you must do either, please do so quietly.
- Laptop use is discouraged — their use detracts significantly from the benefit of coming to class (wouldn't it have been more fun to spend an hour surfing Instagram at home?) and also provides a distraction for other students. If you must use your laptop, please turn the sound off, type quietly, and sit as far towards the back of the room as possible.
- Recording of the class (audio or video) is forbidden without prior permission from the instructor.

Accommodations for students with disabilities. If you have a disability and have an accommodations letter from the Disability Resources office, we encourage you to discuss your accommodations and needs with us as early in the semester as possible. We will work with you to ensure that accommodations are provided as appropriate. If you suspect that you may have a disability and would benefit from accommodations but are not yet registered with the Office of Disability Resources, we encourage you to contact them at `access@andrew.cmu.edu`.

Academic honesty. All class work should be done independently unless explicitly indicated on the assignment handout or in accordance with the general homework policy above. You may *discuss* homework problems with classmates, but must write your solution by yourself. If you do discuss assignments with other classmates, you must supply their names at the top of your homework / source code. No excuses will be accepted for copying others work (from the current or past semesters), and violations will be dealt with harshly. (Getting a bad grade is much preferable to cheating.)

The university's policy on cheating and plagiarism can be found here: `https://www.cmu.edu/policies/student-and-student-life/academic-integrity.html`. In part it reads "In any manner of presentation, it is the responsibility of each student to produce her/his own original academic work." You should be familiar with the policy in its entirety.

In particular: use of a previous semester's answer keys or online solution manuals for graded work is absolutely forbidden. Any use of such material will be dealt with as an academic integrity violation.

Frequently Asked Questions

Is there some extra work I can do to improve my grade? We cannot make exceptions to the coursework and grading policy. If you are concerned about your grade, please see the instructor or the TA ASAP. There will be no exceptions to this policy during or after the class has completed.

I have to be out of town, and I would like an extension on my homework. Can I have one? It is not possible to accommodate individualized deadlines for everyone. You can always turn your homework in early or remotely (if the link is not available, please ask a TA).

4. Statement on Student Well-Being

Take care of yourself. Do your best to maintain a healthy lifestyle this semester by eating well, exercising, avoiding drugs and alcohol, getting enough sleep and taking some time to relax. This will help you achieve your goals and cope with stress.

All of us benefit from support during times of struggle. You are not alone. There are many helpful resources available on campus and an important part of the college experience is learning how to ask for help. Asking for support sooner rather than later is often helpful.

If you or anyone you know experiences any academic stress, difficult life events, or feelings like anxiety or depression, we strongly encourage you to seek support. Counseling and Psychological Services (CaPS) is here to help: call 412-268-2922 and visit their website at `http://www.cmu.edu/counseling/`. Consider reaching out to a friend, faculty or family member you trust for help getting connected to the support that can help.

If you or someone you know is feeling suicidal or in danger of self-harm, call someone immediately, day or night:

CaPS: 412-268-2922

Re:solve Crisis Network: 888-796-8226

If the situation is life threatening, call the police:

On campus: CMU Police: 412-268-2323

Off campus: 911

The course staff of String Algorithms wholeheartedly agrees with the above. The polices set forth in this document are designed to help you have a successful semester in which you learn a lot. If you have questions about this or your coursework, please let us know.