CS 514: Advanced Algorithms II – Sublinear Algorithms (Spring 2020)

Course Syllabus

1 Course Information

General information

Instructor: Sepehr Assadi. Email: `sepehr.assadi@rutgers.edu`. Office: CoRE 310.

Lectures: Tuesdays 12:00pm – 3:00pm at TIL-116 (Livingston campus).

Office hours: Mondays 3:00pm – 4:00pm or by appointment at CoRE 310 (Busch campus).

Prerequisites: Mathematical maturity and basic background on algorithms, complexity theory, discrete mathematics, and probability at the introductory level (e.g., undergraduate courses). Some experience with theoretical computer science beyond the introductory level (say, Advanced Algorithms (CS 513)) will be helpful but it is not necessary.

Textbook: There is no official textbook for this course but some useful resources are listed on the course webpage.

Webpage: https://www.cs.rutgers.edu/~sa1497/courses/cs514-s20.html

The webpage contains updated syllabus information as the semester progresses and a calendar. The lecture notes and problem sets will also be posted on the webpage. We will use Canvas for the announcements and releasing grades.

Students with disabilities: The students with disabilities are encouraged to discuss with the Instructor any appropriate accommodations that we might make on their behalf following the guidelines of the Office of Disability Services[1].

Academic integrity: The students are expected to follow Rutgers academic integrity policy[2] for all their work in this course. Please familiarize yourself with this policy if you have not done so yet.

What should you expect to learn from this course?

The main goal of this course is to present the students with various advanced algorithmic ideas through the lens of sublinear algorithms – these are "ultra efficient" algorithms whose resource requirements (e.g. time or space) are substantially smaller than the size of the input that they operate on, and are typically needed for processing massive datasets. Additionally, this course aims to prepare the interested students for doing research on algorithms in general and sublinear algorithms in particular.

The following is a tentative list of topics that will be covered in this course:

[1] https://ods.rutgers.edu/my-accommodations
[2] http://academicintegrity.rutgers.edu/academic-integrity-at-rutgers/

- **Sublinear Time Algorithms:** Which problems can be solved in time faster than even reading the entire input once? We will cover sublinear time algorithms for property testing, distribution testing, and graph problems. We will also examine query complexity as a main tool for proving lower bound on the performance of sublinear time algorithms.
- **Streaming Algorithms:** Which problems can be solved in space smaller than what is needed to store the entire input? We will cover streaming algorithms for statistical estimation, numerical linear algebra, and graph problems. We will also examine communication complexity as a main tool for proving lower bound on the performance of streaming algorithms.

Along the way, we will learn about various key ideas such as probabilistic analysis of algorithms, compressed sensing, dimensionality reduction, sparsification, sketching, (composable) coresets, etc., that are used extensively in algorithm design as a whole and sublinear algorithms in particular.

2 Assignments and Grading

Grading

- 40% Problem sets
- 40% Project
- 20% Scribe notes and participation

Problem Sets

– **Timing:** There will be *three* problem sets in this course and a tentative schedule of release and due dates are available on the course calendar. Problem sets will be released on a Tuesday and are **due three weeks** later by **11:59 pm on Monday**.

– **Format:** Problem sets should be turned in on Canvas as **a single pdf file** containing the solutions in order. Moreover, **solutions must be typeset in LaTeX**. Simple instructions on using LaTeX are available on the course webpage and a template will be released with each problem set.

– **Collaboration:** Problem sets can (and probably should) be done in teams of up to three students. However, (1) the students should write their solutions *completely* independently (in particular, you should understand and be able to explain everything that is written in your solution); (2) you should include the name of your collaborators in your solutions.

When writing your solutions, you are *allowed* to use materials not discussed in the class (say, related research papers) **as long as you cite these references appropriately**. However, you are *not* allowed to get help from someone who is not currently enrolled in the class.

– **Difficulty of the problem sets?** The problem sets are going to be considerably challenging. You have three weeks to attempt each problem set so plan your time accordingly. However, do not get discouraged if you cannot solve some of the problems as they can be really hard – the goal of these problems is to familiarize you with the type of technical problems that appear typically in research on algorithms, and, exactly as in research, you should not expect to be able to solve every problem.

Project

– There is a final project that will consist of exploring a topic of interest related to this course. This particularly involves reading one or two recent research papers **in complete details** to get a sense of the background on a research problem and then exploring ideas for addressing this problem.

– The students will have a presentation of the problem, its background, and their progress to the class at the end of the course. You will also be asked to present a written report of the project, focusing primarily on *your understanding* of the problem and any progress you have made by then.

– More details on the project plus the list of potential project ideas will be released later in the semester. However, feel free to discuss any project idea you may have with the Instructor before that.

Scribe notes and participation

– For each lecture, there will be one team (of one or two students) in charge of taking detailed notes, typing them in LaTeX, preparing any needed figures, and sending them to the Instructor **by 11:59pm on Friday** after the lecture. These notes will be posted on the course website.

– If the quality of the notes are not satisfactory (e.g., having incomplete or imprecise proofs, many typos, lack of proper illustrations and examples, etc.), you will be asked to revise the notes until we converge to a version that is useful for your classmates. Keep in mind that the target audience of these notes is you – students – a couple of weeks after the lecture, when you already have forgotten what it was about. As such, every detail of the lecture should be addressed in the notes.

– The assignments of scribe notes will be fixed in a week or two – the first couple of lectures are scribed by the Instructor as an example. A LaTeX template with proper instructions for scribing the notes will be released as well.

– You are expected to participate in every lecture but definitely for the one you are scribing for. Further class participation (e.g., interaction during the lecture, asking good questions) can also (positively) influence your grade and is strongly encouraged.