DS-563 / CS-543: Algorithmic Techniques for Taming Big Data

Boston University
instructor: Krzysztof Onak
website: https://onak.pl/ds563

Spring 2025

Course Description: Growing amounts of available data lead to significant challenges in processing them efficiently. In many cases, it is no longer possible to design feasible algorithms that can freely access the entire data set. Instead of that, we often have to resort to techniques that allow for reducing the amount of data such as sampling, sketching, dimensionality reduction, and core sets. Apart from these approaches, the course will also explore scenarios in which large data sets are distributed across several machines or even geographical locations and the goal is to design efficient communication protocols or MapReduce algorithms.

The course will include a final project and programming assignments in which we will explore the performance of our techniques when applied to publicly available data sets.

Instructor: Krzysztof Onak (except for sensitive topics, please use Piazza to contact the instructor)

Meetings:

- **Tuesdays:** 3:30–4:45pm, CDS 701
- **Wednesdays:** 1:25–2:15pm, SOC B61 or 2:30–3:20pm, CGS 423
- **Thursdays:** 3:30–4:45pm, CDS 701

Tentative Office Hours: We will run survey in class and/or on Piazza to find best possible office hour slot(s). For now (say, the first week), we aim for the following slots:

- **Tuesday:** 1–2pm, CDS 1443
- **Wednesday:** 4–5pm, CDS 1443
- (or by appointment)

Learning Objectives

- **Big Data Techniques:** In the classic computing setting, the computer has unconstrained access to the entire data set while it computes a solution to a given problem. The main goal of this course is to develop various ways of thinking about processing big data that address scenarios in which this data access is constrained due to significantly limited computational resources. The content of the course

is divided into a few parts, each devoted to a different approach to processing big data. Within each part, we will explore a number of computational scenarios and corresponding algorithms. We will also mention or explore the limitations of many of the techniques.

- **Developing mathematical toolkit:** We will develop and learn a number of mathematical tools, which will be used for analyzing algorithms and developing lower bounds for them. In particular, since many big data algorithms involve randomness, we will introduce many tools for analyzing random variables.
- **Rigorous analysis of algorithms:** Lectures, discussion sections, and homework will encourage students to learn and practice the art of rigorous analysis of algorithms. This involves both using mathematical proofs to prove the correctness and efficiency of algorithm, and constructing counterexamples for failed attempts.
- **Turning theoretical algorithms into working implementations:** We will discuss and explore ways of converting theoretical ideas and algorithms with provable guarantees into efficient implementations. For instance, many theoretical analyses introduce impractically large constant factors. To address this, we will consider running experiments to determine much smaller and more practical constants.

Course Information and Tools

- Course website: https://onak.pl/ds563 (or https://onak.pl/cs543)
- Piazza (announcements and discussions): https://piazza.com/bu/spring2025/ds563cs543
- Homework, project proposal, and final project report are to be submitted via Gradescope

Prerequisites

- **Discrete mathematics, logic, and proofs:** Familiarity with basic discrete mathematics is required. Knowledge of basic logic and ability to conduct mathematical proofs are required. These topics are covered in the DS math sequence (DS-120, DS-121, DS-122) and CS-131, as well as many discrete math textbooks, including Stein, Drysdale, Bogart "Discrete Mathematics for Computer Scientist."
- **Algorithms:** Familiarity with basic topics in algorithms and computation complexity (commensurate with DS-320, CS-330, EC-330, or equivalent) is required. These topics are covered in many textbooks, including Cormen, Leiserson, Rivest, Stein "Introduction to Algorithms," and Kleinberg, Tardos "Algorithm Design."
- **Linear algebra, probability, and calculus:** Familiarity with basics of linear algebra, probability, and calculus is required. These topics are covered in the DS math sequence (DS-120, DS-121, DS-122) and multiple other courses across the BU campus (including CS-132, CS-237, MA-115, MA-242, EK-381). Some ot these topics are covered in textbooks such as Strang "Introduction to Linear Algebra," Lay, Lay, McDonald "Linear Algebra and Its Applications," and Pishro-Nik "Introduction to Probability, Statistics, and Random Processes."
- **Programming:** Fluency with programming and basic data structures is required (commensurate with DS-110, CS-111, EK-125, or equivalent). Any programming language can be used for programming assignments but recommended suitable programming languages include Python, C++, Java, and Rust.

Self–Assessment Questionnaire. Since advanced courses offered through the Faculty of Computing & Data Sciences are meant to be open to students from various disciplines, we provide a questionnaire to assist students with self–assessment and placement. See the appendix.

Course Requirements

Apart from active participation, the the course requirements include theoretical homework problem sets, three experimental programming assignments, and a final project. The overall grade will be based on the following factors:

- class participation: 7.5%
- theoretical homework: 22.5%
- programming assignments: 22.5%
- project proposal: 7.5%
- final project: 40%

Class participation. The course requires active class participation. It is important to attend all meetings (lectures and discussion sections). If you miss a meeting, please talk to other students about any missed material. It is also highly recommended to come to office hours to discuss any material that one finds challenging and to actively participate in Piazza discussions.

To encourage active participation, we will keep track of virtual participation cards. When you participate in good faith during class, we will collect your card. Every now and then (probably roughly every two weeks), we will announce that the cards have been distributed back. We will keep track of the number of times the participation card was collected for each student on Gradescope. Your credit for class participation will be a (possibly non-linear) function of the number of times your card was collected and may include other factors as well.

The class participation grade contribution includes 2% for timely signing up for a final project presentation slot.

Programming assignments. The course will feature two programming assignments in which students will implement algorithms covered in class and apply them to data sets of their choice. Collaboration here is not allowed (except for discussing high–level ideas), i.e., students are required to implement algorithms and run experiments on their own.

Final project. Possible final projects ideas include but are not limited to

- implementing an algorithm not covered in class and testing its practical performance on real–world data,
- creating an open–source implementation of one of the algorithms with easy to follow documentation,
- developing a new algorithm with good theoretical or practical guarantees.

The outcome of a project will be a short technical report, describing obtained results and conclusions. As opposed to programming assignments, students are allowed to work in teams of 2 or 3. A list of potential projects topics will be provided, but students are encouraged to develop their own ideas. These projects have to be approved by the instructor.

LaTeX. All submitted materials and solutions have to be typed. It is strongly suggested to use LaTeX.

Late submissions. You may submit your homework one day late, but your grade may be reduced by 10%.

Grade cutoffs. I will determine grade cutoffs after all assignments and exams have been graded. Grade cutoffs will take into account my assessment of the difficulty level of the assignments and exams, and my assessment of what is expected for each letter grade.

Tentative schedule of assignments:

- Homework 1 (theory/math): due Feb 5
- Homework 2 (programming): due Feb 19
- Project Proposal: due March 7
- Homework 3 (theory/math): due Mar 19
- Homework 4 (programming): due Apr 2
- Homework 5 (theory/math): due Apr 16
- Final project presentations: April 29–May 1
- Final Project Report: due May 8

Code of conduct

Homework collaboration policy. You are allowed to collaborate on your homework with up to three of your classmates. However the assignments you hand in should be written up by yourself and represent your own work and thoughts. In particular, you are allowed to discuss ideas with them in person, but as a rough rule, nobody should leave the room with anything written down. If you really understand the discussion, you should be able to reconstruct it on your own.

Your must list your collaborator's names on the top of your assignment. If you don't work with anyone, you must write "Collaborators: none."

Academic code of conduct. You have to adhere to BU's academic conduct policy:
https://www.bu.edu/academics/policies/academic-conduct-code/
Additionally, this webpage has great examples of what is and what is not acceptable:
https://www.bu.edu/cs/undergraduate/undergraduate-life/academic-integrity/

Generative AI. Using generative AI tools such as ChatGPT or Google Bard is not allowed for homework, including both theory and programming assignments, with the exception of looking up syntax of the programming language you are using.

It is allowed to use it as a search tool to learn more about subjects being covered. If you use it for your final project, you have to strictly delineate what was contribution was and what was created created using generative tools. The main ideas and conclusions from your project have to be yours and you have to be able to defend them. You also are personally responsible for everything delivered.

See also the CDS GAIA policy:

https://www.bu.edu/cds-faculty/culture-community/gaia-policy/

Materials

There is no textbook. A good list of resources on many of the topics covered in this class—including books, surveys, lectures notes, and presentations—can be found at

https://sublinear.info/index.php?title=Resources

Lecture recording

This course will not provide lecture or discussion section recordings. However, we may allow some students to record lectures as a disability accommodation. Sharing these recordings without permission of class participants is not allowed and they should be deleted when the course completes.

If this lecture recording policy makes you uncomfortable, please discuss it with the instructor.

Reasonable Accommodations

If you are a student with a disability or believe you might have a disability that requires accommodation, please contact the Office for Disability Services at 617-353-3658 or access@bu.edu. Please also notify the instructor about any accommodation that you may require as soon as possible. We may not be able to provide some accommodations if we do not learn about them sufficiently early.

Electronic Devices in Class

tl;dr: No.

Using laptops, cellphones, tablets, and other similar electronic devices is generally not allowed. If you want to use your laptop or tablet for taking notes, it is allowed but you may be asked to show your notes to the lecturer after the lecture. Even if you use your device for making notes, you are not allowed to use it for other purposes, such as replying to emails or browsing the web.

Lectures vs. Discussion Sections

We will treat what is officially a lecture and discussion section equally. Smaller meetings (i.e., officially discussion sections) allow for more interaction, but sometimes we will simply continue the topic that was discussed in with topic that was discussed in a larger meeting.

Tentative List of Topics

Actual topics to be covered may change, but the plan is to cover five main approaches to processing big data. For each, we will explore both the related computation model (or models) and go over a number of algorithms.

- **Section 1: Data Projections (Including Streaming Algorithms)**
  - Frequency Estimation and Count-Min Sketch.
  - Estimating the Number of Distinct Elements.
  - Frequency Moments. The AMS algorithm for F_2.
  - Compressed graph representations with applications (graph sketches).
  - Dimensionality reduction and the Johnson–Lindenstrauss lemma.
  - Nearest neighbor search via Locality Sensitive Hashing.
- **Section 2: Selection of Representative Subsets**
  - Misra–Gries summaries.
  - Coresets for Quantiles.
  - Coresets for k–median.
- **Section 3: Distributed Computation**
  - Massively Parallel Computation Model.
  - Sorting on MPC.
  - Computing PageRank on MPC.
  - Computing large matchings on MPC.
- **Section 4: Sampling from Probability Distributions**
  - Learning discrete distributions.
  - Uniformity testing.
  - Better algorithms for monotone and unimodal distributions.
- **Section 5: Querying and Sampling Subsets of Data Sets**
  - Estimating the minimum vertex cover size.
  - Testing the monotonicity of a sequence.
  - Estimating the number of edges in a graph.

Appendix: Self–Assessment Questionnaire

Programming:

- Do you know how to write simple programs?
- Can you write a program that reads a database of dates of birth and salaries represented as a text file, in which each line is of the form "name;date of birth;salary"? The program should create an appropriate representation of the database in memory. Sample input file:

```
Alice;1991-09-01;$123456.78
Bob;1993-09-03;$98765.43
```

- Can you write a program that simulates tossing an unbiased coin 10,000 times and reports 10 most common subsequences of five consecutive coin tosses?

Algorithms:

- What is binary search?
- What are (balanced) binary search trees?
- How much time does it take to sort a sequence of n real numbers?
- What is Depth–First Search and Breadth–First Search?
- What are hash tables?

Mathematics:

- Can you conduct simple mathematical proofs?
- Can you prove that
$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$
for all positive integers n?
- What is the rank of a matrix? What is the rank of this matrix?
$$\begin{pmatrix} 3 & 2 & -1 & 5 & 3 \\ 0 & 11 & 3 & 1 & 1 \\ -6 & 7 & 5 & -9 & -5 \\ 0 & 9 & 4 & -4 & 0 \end{pmatrix}$$
- What is the dot product of two vectors?
- Do you know (and can prove) Markov's inequality?
- Do you know (and can prove) Chebyshev's inequality?
- If X and Y are random real variables, when does $E[X+Y] = E[X] + E[Y]$?
- If X and Y are random real variables, when does $E[X \cdot Y] = E[X] \cdot E[Y]$?