DS-320: Algorithms for Data Science

Boston University

Fall 2023

Instructor: Prof. Krzysztof Onak (konak@bu.edu)
Office Hours: Wednesday 11am–1pm, CDS 1443

Teaching Fellow: Dragos Ristache (dragosr@bu.edu)
Office Hours: Monday 11am–1pm, blue northwest corner of CDS's 13th floor

Lectures: Tuesday and Thursday 1:30–2:45pm, CGS 527
Discussion Section A2: Monday 3:35–4:25pm, CAS 220
Discussion Section A3: Monday 4:40–5:30pm, CDS 263

Official course description: This course covers the fundamental principles underlying the design and analysis of algorithms. We will walk through classical design methods, such as greedy algorithms, design and conquer, and dynamic programming, focusing on applications in data science. We will also study algorithmic methods more specific to data science and machine learning. The course places a particular emphasis on algorithmic efficiency, crucial with large and/or streaming data sets, for which multiple scans of data are infeasible, including the use of approximation and randomized algorithms.

Learning objectives:

- creative problem solving and algorithmic thinking,
- the ability to identify and formally describe an algorithmic problem underlying a given applied problem,
- learning a repertoire of algorithmic techniques and learning how to apply them to real world problems,
- learning how to read and write formal definitions and statements,
- writing clear and convincing arguments.

BU Hub: Quantitative Reasoning II and Critical Thinking

Prerequisites

Please consult the official data science program prerequisites (there have been some changes I believe) for up to date version if you are a data science major.

This is a theoretical problem–solving and proof–writing course. Whatever background you are coming from, your preparation should include:

- Basic programming: We won't directly program in this class, but understanding what programming is and the ability to read and write pseudocode is crucial.
- Math: Basic calculus, combinatorics, probability, and linear algebra are all very useful. A basic understanding what constitutes a proof and the ability to write simple proofs is very useful. This is something that we will practice and improve on in this class. Any more advanced class that gives you training in this can be very useful.

Course information and tools

- Course website: `https://onak.pl/ds320`
- Piazza (announcements and discussions): `https://piazza.com/bu/fall2023/ds320/home` (code: 032SD)
- Gradescope (homeworks): TBD (see the website for updates)

Course Requirements

The final grade will be a combination of the following factors:

- class participation (in particular, asking and answering questions in class and on Piazza): 5%
- weekly problem sets: 38% overall
- two midterms: 19% each
- final: 19%

Weekly Problem Sets

- Expect to spend 10 hours per week on homework.
- **Late policy:** Homeworks will typically be due on Wednesdays at 11:59pm. You can submit your solution up to one day late (typically 11:59pm on Thursday) but you may lose 10% of credit for this.
- Your lowest homework score will be dropped at the end of the semester.
- Your written assignments must be prepared with LaTeX (or other software with the same functionality), not handwritten. You may use Overleaf.
- You must submit your homework via Gradescope in the PDF format.

- Regrade requests must be submitted within 7 days of receiving the graded assignment and only via Gradescope. You must submit an explanation which problems were graded incorrectly and an argument why the solution is in fact correct. If you submit a regrade request, you accept that the entire assignment or exam may be regraded, not just the problem that you believe was graded incorrectly.

Midterms and the final exam. The course will have two in-class midterms (Lecture 10 and Lecture 18) and a final exam (to be scheduled, during the finals period). They will all be closed book.

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/`

Generative AI. Using generative AI tools such as ChatGPT or Google Bard is not allowed for this class. See also the CDS GAIA policy:
`https://www.bu.edu/cds-faculty/culture-community/gaia-policy/`

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.

Lecture recording

In this course, your voice and image maybe be recorded by the instructor. In particular, the plan is to share video recordings of lectures with the class participants for instructional purposes. These recordings are not to be shared outside of the classroom without permission of persons being recorded. The recordings may be retained later to improve the quality of future educational offerings but they will not be posted publicly.

Additionally, some students may be allowed to record lectures as a disability accommodation. Sharing these recordings without permission of class participants is not allowed.

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

Materials

There is no required textbook. There will be suggested readings from various textbooks and lecture notes during the course. If you prefer to use an algorithmic textbook, two popular books are "Algorithm Design" by Kleinberg and Tardos and "Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein.

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.

Tentative Schedule

- Lecture 1: Overview and policies. Runtime and asymptotics.
- Lecture 2: Insertion Sort, induction, and the comparison–based lower bound.
- Lecture 3: Abstract data types and Depth–First Search (DFS).
- Lecture 4: More DFS, including Topological Sort.
- Lecture 5: Breadth–First Search (BFS) and testing bipartiteness.
- Lecture 6: Greedy algorithms I: Dijkstra's algorithm.
- Lecture 7: Greedy algorithms II: optimal caching.
- Lecture 8: Greedy algorithms III: scheduling.
- Lecture 9: Greedy algorithms IV: Huffman codes.
- Lecture 10: Midterm.
- Lecture 11: Divide & Conquer I: Mergesort and solving recurrences.
- Lecture 12: Divide & Conquer II: closest pair of points.
- Lecture 13: Divide & Conquer III: integer and matrix multiplication.
- Lecture 14: Dynamic Programming I: weighted interval scheduling.
- Lecture 15: Dynamic Programming II: segmented least squares.
- Lecture 16: Dynamic Programming III: knapsack.
- Lecture 17: Dynamic Programming III: Bellman–Ford algorithm.
- Lecture 18: Midterm.
- Lecture 19: NP–completeness.
- Lecture 20: Linear Programs I: introduction.
- Lecture 21: Linear Programs II: examples and duality.
- Lecture 22: Linear Programs III: more duality.

- Lecture 23: Zero sum games and the minimax theorem.
- Lecture 24: Multiplicative weight update.
- Lecture 25: Stable matching.
- Lecture 26: Approximation algorithms: random and online.
- Lecture 27: Sample streaming and big data algorithms.