DSCI 512: Algorithms and Data Structures

Contents

Welcome to DSCI 512! This course introduces fundamental data structures and algorithms, complexity analysis, and techniques for improving Python code performance, with a focus on solving data science problems.

Course Learning Outcomes

By the end of the course, you will be able to:

1. Analyze the efficiency and trade-offs of fundamental algorithms and data structures using Big-O notation.
2. Select and implement appropriate data structures, such as hash tables or graphs, given a data set.
3. Work with vectors, matrices, tensors, and sparse matrices for computational tasks.
4. Read and interpret recursive functions.

5. Use dynamic programming to solve optimization problems with overlapping subproblems.
6. Model real-world problems as discrete optimization tasks and solve them using Python libraries.
7. Diagnose performance bottlenecks in Python code and explore strategies to improve execution speed.

Teaching Team

Role	Name	Section
Instructor	Vincent Liu	1
Instructor	Hedayat Zarkoob	2
Instructor	Jian Zhu	CL
Lab Instructor	Jungyeul Park	CL
Teaching Assistant	Cindy Zhang	1
Teaching Assistant	Eric Lee	1
Teaching Assistant	Tony Fong	1
Teaching Assistant	Armin Saadat	2
Teaching Assistant	Jordan Yu	2
Teaching Assistant	Julian Becerra	2
Teaching Assistant	Changbing Yang	CL
Teaching Assistant	Mingcong Li	CL

Schedule

Lectures

#	Topic	Lecture Notes	Recommended Readings
1	Time & space complexity, and big O notation	lecture 1	Introduction to Algorithms Chapter 2.2 & 3.1 & 3.2, Runestone Chapter 3
2	Basic data structures	lecture 2	Introduction to Algorithms Chapter 10 & 11.2 & 11.3 & 12, Runestone Chapter 4 & 6.5 & 7.11
3	Vectors, matrices, and tensors	lecture 3	Linear Algebra for Data Science Chapter 5, A Visual Intro to NumPy and Data Representation
4	Sparse matrices	lecture 4	Introduction to PyTorch, FP64, FP32, FP16, BFLOAT16, TF32, and other members of the ZOO
5	Graph algorithms	lecture 5	Introduction to Algorithms Chapter 20, Runestone Chapter 8
6	Recursive algorithms	lecture 6	Introduction to Algorithms Chapter 4, Runestone Chapter 5 and Chapter 7
7	Dynamic programming	lecture 7	Introduction to Algorithms Chapter 14, Dynamic programming
8	Discrete optimization	lecture 8	Linear programming and discrete optimization with Python using PuLP

Labs

Most of the lab session time will be allocated to working through the lab assignment, on your own or in groups. There will be a lot of opportunity for discussion and getting help during the

lab sessions.

Office Hour

Refer to the MDS calendar.

Deliverables

Lab Assignments

There will be one lab assignment per week.

- Labs are submitted via Gradescope.
- Each lab is worth 12.5% of your final grade (total of 50%).

Quizzes

There will be two quizzes, as scheduled on the MDS Calendar.

- Each quiz is worth 25% of your final grade (total of 50%).

Installation

The course's conda environment file is here.

- To create the environment: `conda env create -f dsci512env.yml` (you only need to do this once).
- To activate the environment: `conda activate dsci512env` (you need this for each new terminal window).

In short, run `conda install -c conda-forge nb_conda_kernels` in your **base** environment (this only has to be done once across all courses, so you may have done it already), then launch Jupyter Lab from your **base** environment, and finally select the `dsci512env` kernel from within Jupyter.

Resources

Books

- Introduction to Algorithms, by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein.
- Data Structures and Algorithms in Python, by Goodrich, Tamassia and Goldwasser.
- Algorithms, available here, by Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani.

Online Reference Material

- [runestone] Problem Solving with Algorithms and Data Structures using Python
- 500 Data Structures and Algorithms practice problems and their solutions
- Recursion practice problems
- P vs. NP and the Computational Complexity Zoo (video)

Use of LLMs

LLMs, such as ChatGPT, can be helpful tools if we use them responsibly. In this course, students are permitted to use these tools to gather more information, review concepts, or brainstorm, and students must cite these tools if they use them for assignment. Having said all this, it is not permitted to write any given assignment via copying and pasting AI-generated responses.

Attribution

The course is built upon previous years' materials developed by previous instructors.

Policies

Please see the general MDS Policies.