CS 277 : Algorithms and Data Structures for Data Science
Syllabus

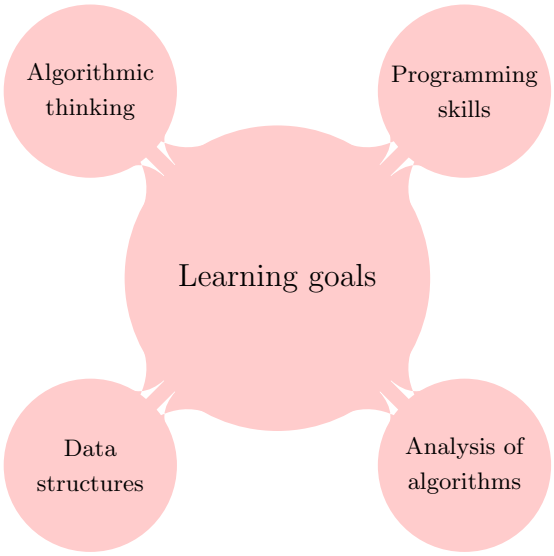
Instructors: Mahesh Viswanathan, Harsha Srimath Tirumala

Overview

This is a course on algorithms and data structures. We will explore different algorithm design paradigms and design algorithms to solve a variety of computational problems. Some examples include finding the cheapest flight route, optimal search through a structured dataset, stock trading, finding the distance between users of social networks, building a course schedule planner, setting up recommendation systems and compression of large data. At the end of this course, students will develop the skill to model problems as computational problems and write programs to solve them, analyze their resource bounds and justify their correctness.

Learning Goals

- Algorithmic thinking - Students will be able to model computational problems and design algorithms to solve problems efficiently.
- Programming skills - Students will develop programming skills to write programs in Python to solve computational problems.
- Analysis of Algorithms - Students will be able to analyze algorithms and give asymptotic resource bounds (time, space) and judge algorithms' correctness for computational tasks.
- Data Structures - Students will learn the inner workings of various data structures and develop the ability to pick appropriate data structures to solve computational problems.
- Data Science - Students will be able to apply clustering and sampling algorithms to model and solve relevant data science problems



Learning Objectives

- Asymptotics of functions : Comparing functions asymptotically, verifying asymptotic relationships among functions
- Recurrence relations : Interpreting and Writing recurrences to model resource usage, Solving recurrences to give asymptotic bounds
- Algorithm design : Design algorithms using paradigms such as Recursion, Divide and Conquer, Greedy, Dynamic programming, Linear programming
- Analysis : Analyze algorithms for resource usage, correctness
- Programming skills : Writing python programs, Debugging code, Use of libraries/code bases, Optimizing code
- (In)Efficiency : Usage of Non-determinism in algorithms, NP, P vs NP and reductions among Computational problems
- Data Science : Design Clustering/Sampling based algorithms to solve data science problems
- Data Structures : Understand various ways of storing data and selecting the best fit for meeting problem requirements
- Graphs : Structure, Implementation, Search, Shortest paths, algorithms for other computational problems

Assessments

This course will involve lab assignments, homework assignments and exams.

Lab

Lab assignments are designed primarily to develop programming skills and problem solving ability. Lab assignments will typically have 3 programming questions for students to work on. These questions will be based on content covered during the same week in lectures. Every friday[1] we will preview the lab assignment in class and provide appropriate context for the questions in consideration. Students can use the remaining class time as open office hours for the lab assignment of that week.

- Frequency - Weekly (11 in total)
- Type - Individual submission
- Grade contribution - 30% of total score
- Late submissions - 90% credit for submissions made within 24 hours of deadline
- One free lab extension - Each student will have a one week extension for a lab assignment of their choice. In order to receive this, please fill and submit the following google form within 24 hours of the deadline of the chosen lab - Lab Extension form

[1] except for exam review fridays - February 20 and March 27

Homework

Homework assignments are designed to develop algorithmic thinking, design and analysis of algorithms. Students will typically work on 3 problems - they must understand the requirements of each problem and then design algorithms to solve them. Students will be expected to analyze resource usage (time and space complexities) and justify correctness of algorithms.

- Frequency - Bi-Weekly (6 in total)
- Type - Group (of up to 3 members) submissions[2]
- Grade contribution - 25% of total score
- Late submissions - 90% credit for submissions made within 24 hours of deadline
- One free homework extension - Each student will have a one week extension for a homework assignment of their choice. In order to receive this, please fill and submit the following google form within 24 hours of the deadline of the chosen homework - Homework Extension form

Exams

Exams are designed to test students' problem solving ability and consist of some multiple choice questions and a few problems on algorithms. Exams will be held at the CBTF and students are expected to sign up for slots to take their tests.

- Frequency - 3 in total
- Type - Individual assessments at CBTF
- Grade contribution - 45% of total score (15% for each of three exams)
- Second chance - Students will have the option of retaking one of exam 1 or exam 2 during their final exam CBTF appointment.

Exam schedule :

- Exam 1: Thursday February 26 to Sunday March 1 on CBTF.
- Exam 2: Thursday April 2 to Sunday April 5 on CBTF.
- Final exam: Friday May 7 to Thursday May 14 on CBTF.

Office hours

The course office hours schedule can be found here. In addition, you can write to me at harshast@illinois.edu to set up a meeting to discuss anything you want to. Feel free to drop by my office at Siebel 2322 as well - it is always a pleasure to meet students!

[2]students can submit different homework in different groups

Expected behavior

Expectations		
Aspect	Students	Staff
Class participation	Attention to concepts presented	Passionate and engaging lectures
Class interactions	Non-disruptive (student) interactions, ask questions	Positive attitude to questions, seek feedback
Community	Discuss assessments with peers **AFTER** working individually	Promote exchange of ideas by allocating time
Discipline	Submit assessments on time	Share assessments with clear expectations
Grading	Go through grading rubric	Fair, objective grading based on grading rubric
Preparation	Read suggested reading before/after class	Share course material in advance
Work	Earnest work following guidelines (NO AI)	Strong commitment to student learning