Welcome to CSE 565!

Instructors:

Antonio Blanca
Ph.D. in CS (2016)

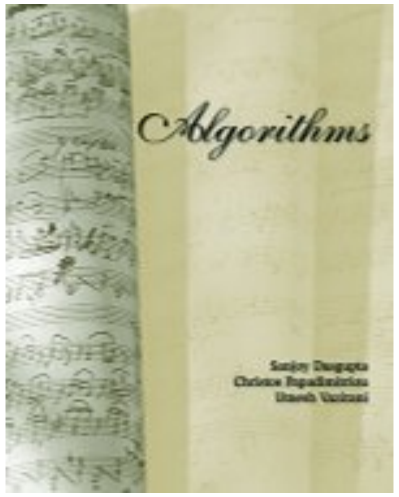
Research: Randomized Algorithms

Website: https://www.cse.psu.edu/~azb1015/

Office hours: Mondays 4:15 to 5:15 pm (Zoom)

Guneykan Ozgul (TA)
Ph.D. student
Office hours: Zoom, TBA

Course Info:



Textbook: Algorithms (Dasgupta, Papadimitriou and Vazirani)

Class Resources:

- Gradescope (for turning in your homework)
- Campuswire for **all** questions
- Canvas (for everything else)
- Plan to provide all my notes in advance.

Additional (recommended) textbook: Algorithm Design (Tardos and Kleinberg)

Homeworks:

- Problem set posted every other Friday
- Due the following Friday at Midnight
- Submit through Gradescope **(Class code: later)**
- Homework accounts for 20% of your grade
- We will drop the HW with the lowest score, so **no exceptions on deadlines**
- "I'll take 20%" option: if you don't arrive to a solution, you can opt for this
- We will grade a subset of the problems and post solutions to all
- **Check the syllabus for important details** (e.g., submitting/formatting)

Collaboration:

- You are welcome to work on your own or in groups of up to 3 students
- **Every student most write her/his own solutions in her/his own words!**
- All collaborators & references should be acknowledged

Exams:

- Two midterms - 20% of your grade each
  - Midterm 1 - In class, TBA
  - Midterm 2 - In class, TBA
- One cumulative Final Exam - 30% of your grade

Getting help QA:

- Homework questions? Campuswire
- ...other questions? Campuswire
- Course logistics? Campuswire
- Re-grading? Gradescope
- Contacting the Instructors? Campuswire or office hours

Email responses from the Instructors may be slow

Prerequisites:

- CMPSC 465 or an equivalent theoretical course on Algorithms and Data Structures

Some things you should know:

- **Reading/writing Proofs** (including induction, contradiction, ...)
- Asymptotic notation (e.g. "big-O")
- Elementary data structures: lists, stacks, queues, sorted arrays, balanced trees, heaps
- Graphs and trees algorithms: DFS, BFS, shortest paths, MSTs
- Basic design paradigms: divide and conquer, greedy algorithms, dynamic programming, brute force, etc.
- Basic mathematical tools: arithmetic and geometric series, counting permutations, vectors and matrices, etc.

Why study algorithms?

- Analyzing correctness and resource usage
- Feasibility (what can and cannot be done efficiently)
  - NP-completeness
- Successful companies (Google, Mapquest, Akamai)
- Computation is fundamental to understanding the world
  - social networks, brains, black holes, evolution
- **Interviews!**
- **Lots of FUN!**

Course objectives:

1. Classical algorithms
2. Analysis of algorithms: correctness, running time and space/memory utilization
3. Classical design techniques
4. Design algorithms using the building blocks presented in this class
5. Introduction the Complexity Theory
6. Coping with NP-completeness: randomized and approximation algorithms
7. Streaming algorithms (input is too big to store)
8. Quantum Computing

Course objectives:

Theoretical study: **proofs** of correctness and performance

1. Classical algorithms

2. Analysis of algorithms: correctness, running time and space/memory utilization

3. Classical design techniques

4. Design algorithms using the building blocks presented in this class

5. Introduction the Complexity Theory

6. Coping with NP-completeness: randomized and approximation algorithms

7. Streaming algorithms (input is to big to store)

8. Quantum Computing