# B403 Introduction to Algorithm Design and Analysis

## §1 Introduction

Qin Zhang

**Brief self-introduction:**

My name: Qin[Chin] Zhang

I’ve worked on theory/algorithms for 17+ years;

My main interest is *Algorithms for Big Data*, *Theoretical Foundations of Machine Learning*, and recently *Quantum Computing*

I’ve published extensively in all top conferences/journals in theory/algorithms

I write experimental papers too, and have published in all top databases, data mining and machine learning venues

## Today's agenda

1. An introduction of the course
2. A touch-base quiz (30 mins)
3. A brief discussion on the solutions

## Course topics

1 : **Introduction**

– Big-O notations, common running times

2 : **Graph Algorithms**

– BFS, DFS, DAG, topological sorting

3 : **Greedy Algorithms**

– Interval scheduling, MST, shortest path

General techniques

4 : **Divide & Comquer**

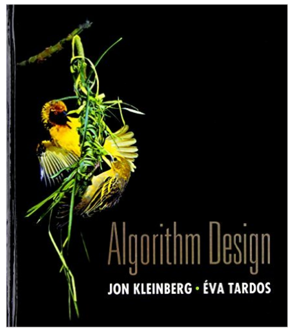
– Mergesort, counting inversions, closest pair

5 : **Dynamic Programming**

– Weighted interval scheduling, subset-sum, edit distance

## Textbooks

- Required textbook
  - Algorithm Design
    by J. Kleinberg and E. Tardos
    Pearson Education



The book comes with slides:

`http://www.cs.princeton.edu/~wayne/kleinberg-tardos/`
(or Google "Algorithm Design slides")

We will NOT use these slides in this course though

## Resources

- **Course website**
  `http://homes.sice.indiana.edu/qzhangcs/B403-23-spring-algorithm/`
  - Various information (e.g., office hours, exam dates)

- **Canvas**
  1. Posting homework assignments and solutions; collecting homework
  2. Announcements
  3. Course lecture notes

## Instructors

- Instructor: Qin Zhang
  Email: qzhangcs@indiana.edu
  Office hours: Wednesday 2-3pm @ Luddy 3044

- AI: Kaiwen Liu
  Email: kaiwliu@iu.edu
  Office hour: Tuesday 4:30-5:30pm, Luddy 2040 area

## Grading

- Attendance 10%

  Conducting five random attendance checks: 0 or 1 absence: 10 points; 2 absences: 8 points; 3 absences: 5 points; 4 absences: 2 points; 5 absences: 0 points.

  If you are sick (w/doctor's note), please inform AI in advance.

- Assignments 30%: Five written assignments

  **The answers should be submitted in a PDF file via Canvas. No extensions will be granted except in cases of illness or emergencies**.

  (One day late, deduct 10% of the points; two days late, deduct 30%; more than two days late, deduct 100%).

- Exams 60%: Mid-term 25%, Final 35%.

- **The final grades will be curved**

## More practice

**Practice** is very important to master algorithm design.

1. Subsections in the textbook that we do not cover in class
2. Solved exercises in the textbook
3. Other exercises in the textbook (do not appear in homeworks).
   Feel free to ask us questions if you meet any difficulty
   (email us the question first so that we can get prepared).
4. Any other questions that you can find online
   – there are tons of algorithm design questions online.
   Again, feel free to ask us questions if you meet any difficulty.

## Prerequisites

Participants must have a background in **math analysis**, **discrete math** and **data structures**,
and have taken

1. C241 Discrete Structures for Computer Science
2. C343 Data Structures

   `https://iu.instructure.com/courses/1560867/pages/schedule?module_item_id=14976212`
3. MATH-M 216 "Analytic Geometry and Calculus II" (or MATH-M 212 CALCULUS II)

## Who should/shouldn't take this course

### Who should take this course

- Has a solid foundation in discrete math (include graph theory, proof by induction/contradiction, etc.), calculus, and data structures
- Wants to be challenged
- Wants to prepare for interviews related to algorithms
- In the "Foundation" track

## Who should/shouldn’t take this course

### Who should take this course

- Has a solid foundation in discrete math (include graph theory, proof by induction/contradiction, etc.), calculus, and data structures
- Wants to be challenged
- Wants to prepare for interviews related to algorithms
- In the “Foundation” track

### Who should NOT take this course

- Finds math proofs difficult, and/or not interested
- Expects assignments/exams are “easy”
- Expects everyone gets ‘A’-level grades
- Has not completed the required prerequisites

## Frequent Asked Questions and Complaints

- Why do you want to make this course so difficult?

  – There seems to be a misconception. My intention is simply to instruct a standard algorithm course using a common textbook, along with assignments and exam questions of moderate difficulty.

## Frequent Asked Questions and Complaints

- Why do you want to make this course so difficult?

  – There seems to be a misconception. My intention is simply to instruct a standard algorithm course using a common textbook, along with assignments and exam questions of moderate difficulty.

- Why this course is (still) a core course in our CS program?

## Frequent Asked Questions and Complaints

- Why do you want to make this course so difficult?

  – There seems to be a misconception. My intention is simply to instruct a standard algorithm course using a common textbook, along with assignments and exam questions of moderate difficulty.

- Why this course is (still) a core course in our CS program?

- Why do we need to do proofs (of the correctness of algos)?

## Frequent Asked Questions and Complaints

- Why do you want to make this course so difficult?

  – There seems to be a misconception. My intention is simply to instruct a standard algorithm course using a common textbook, along with assignments and exam questions of moderate difficulty.

- Why this course is (still) a core course in our CS program?

- Why do we need to do proofs (of the correctness of algos)?

- I can understand most algorithms discussed in class, but those do not help me to solve the homework questions!

## Frequent Asked Questions and Complaints

- Why do you want to make this course so difficult?

  – There seems to be a misconception. My intention is simply to instruct a standard algorithm course using a common textbook, along with assignments and exam questions of moderate difficulty.

- Why this course is (still) a core course in our CS program?

- Why do we need to do proofs (of the correctness of algos)?

- I can understand most algorithms discussed in class, but those do not help me to solve the homework questions!

- I took discrete math many years ago and forgot most of it. Can you provide me with some material so that I can study by myself and try to catch up?

## Thank you!
## Questions?

Next: a touch-base quiz