Introduction

Qin Zhang
Brief self-introduction:

My name: Qin[Chin] Zhang

I’ve worked on algorithm design for 15 years;

My main interest is algorithms for big data, in particular, communication-efficient distributed computation, streaming/sketching algorithms

I’ve published in all top conferences/journals in theoretical computer science (STOC, FOCS, SODA, PODS, SPAA, ICALP; JACM, SICOMP, TALG, Algorithmica).

I write experimental papers too. My papers have been published in premier databases, data mining and machine learning venues (SIGMOD, VLDB, NIPS, ICML, KDD).
Today’s plan
1. An introduction of the course
2. A 40-min touch base exam
3. A brief discussion of the solutions
Courses like *data structure and algorithms*: Implementation and make use of classic data structures and algorithms.

```java
private static void sort(double[] a, int lo, int hi)
{
    if (hi <= lo) return;
    int lt = lo, gt = hi;
    int i = lo;
    while (i <= gt)
    {
        if (a[i] < a[lt]) exch(a, lt++, i++);
        else if (a[i] > a[gt]) exch(a, i, gt--);
        else i++;
    }
    sort(a, lo, lt - 1);
    sort(a, gt + 1, hi);
}
```

**B503: Design and analysis of algorithms.**

\[
\sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{2}{j - i + 1} = 2 \sum_{i=1}^{N} \sum_{j=2}^{N-i+1} \frac{1}{j} \\
\leq 2N \sum_{j=1}^{N} \frac{1}{j} \\
\sim 2N \int_{x=1}^{N} \frac{1}{x} \, dx \\
= 2N \ln N
\]
Course topics

1. **Greedy Algorithms**
   - Interval scheduling, minimum spanning tree, shortest path

2. **Divide & Conquer**
   - Merge sort, counting inversions, closest pairs, fast fourier transform

3. **Dynamic Programming**
   - Weighted interval scheduling, knapsack, sequence alignment, all-pair shortest path

4. **NP and Intractability**
   - Polynomial reduction, NP-completeness, hard problems

5. **Selected topics**
   - Approximation and randomized algorithms
Textbooks

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

  The book comes with slides:
  (or Google “Algorithm Design slides”)
  We do NOT use them in class though.
Resources

- **Course website** (general information)
  http://homes.soic.indiana.edu/qzhangcs/B503-21-spring-algorithm/
  - various information (e.g., office hours)

- **Canvas**
  1. post homework assignments and solutions
  2. homework submissions
  3. announcements
  4. course lecture notes
Instructors

- Instructor: Qin Zhang
  Email: qzhangcs@indiana.edu
  Office hours: Wed. 11am-12pm via Zoom

- Als:
  - Harshad Badiyani (grading, office hour, exercise lectures)

  Helper (not official AI):
  - Yan Song (office hour only)

Emails, office hours: posted on course website
Grading

- Class participation 10%  
  (please turn on the camera of your laptop/mobile devices)
- Assignments 40%: 5 written assignments, each contains 8 questions
  Submit via Canvas before the deadlines.
  Typeset using Latex; if you use Word, make sure that the math notations are easy to read
  No extensions or late homeworks will be granted
- Exams 50%: Mid-term (20%), Final (30%).
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 number 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 **discrete math** and **data structures**, and have taken

1. C241 Discrete Structures for Computer Science
2. C343 Data Structures
   
3. MATH-M 216 ”Analytic Geometry and Calculus II” (or MATH-M 212 CALCULUS II)
Thank you!
Questions?