Introduction

Qin Zhang
Brief self-introduction:

My name: Qin[Chin] Zhang

I’ve worked on algorithm design for 12 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, and have published in all top databases, data mining and machine learning venues (SIGMOD, VLDB, NIPS, ICML, KDD).
Today’s plan
1. An introduction of the course
2. A 45-min touch base exam
Why study algorithms?

Algorithms are used everywhere, any time

**Internet.** Web search, packet routing ...

**Biology.** DNA similarity search, protein folding ...

**Multimedia.** MP3, JPG, face recognition ...

**Social networks.** Recommendations, advertisements ...

**Daily life helpers:** Google maps, auto translation ...

This course tries to introduce some basic concepts, techniques and tools for algorithm design. These may serve as building blocks for solving real-world problems.
It is all about efficiency

We want to *design* algorithms that are *time*, *space* and *communication* efficient.
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.

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

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

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

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

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

- **Selected topics**
  - Approximation and randomized algorithms
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”)
Resources

- **Course website**
  http://homes.soic.indiana.edu/qzhangcs/B503-19-fall-algorithm/
  1. course schedule
  2. 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. 4-5pm @ Luddy 3044

- Als:
  - Haoyu Zhang

Helper (not official AI):
  - Yan Song

Emails, office hours: posted on course website
Grading

• Assignments 30%: 5-6 written assignments
  
  Submit via Canvas before the deadlines.
  Typeset in your favorite software.
  No extensions or late homeworks will be granted.

• Exams 70%: Mid-term 1 (20%), Mid-term 2 (20%), and 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.
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
   
   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)
Thank you! Questions?