Algorithm Design and Analysis

Yuhui Shi
Chair Professor
Room 521 West Wing, South Tower, CoE Building
Department of Computer Science and Engineering
Southern University of Science and Technology
shiyh@sustech.edu.cn

Administrative Stuff

Lecturer: Yuhui Shi

- Tuesday 10:20-12:10/19:00-20:50
- Attendance is expected.

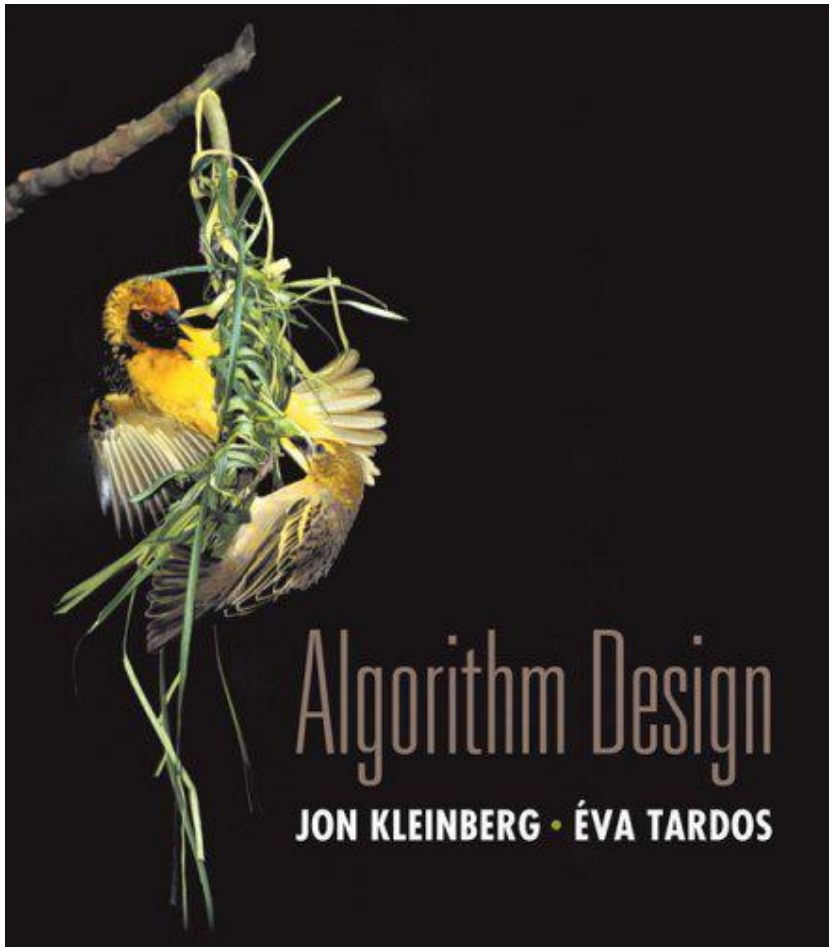
Lab: Yao Zhao & Weiyu Wang

Prerequisite. CS203

Teaching Language: English

Textbook. *Algorithm Design*

by Jon Kleinberg and
Éva Tardos.



Open Office Hours: 14:30 – 16:30AM, on Wednesday of every teaching week
at my office (521 West Wing, South Tower, CoE Building)

Grades

Course grades.

- Final Exam: 40%
- Lab: 30%
- Homework: 20%
- Attendance: 10%
  - We will ask you to sign online during lecture time in several teaching weeks;
  - or We will ask you to sign your names on a piece of paper during lecture time in several teaching weeks;
  - Do not sign for others. If caught, first time, 0 point for the attendance part; second time, 0 point for the whole course;
  - No excuse will be accepted for not attending classes unless I have been notified by an official permission for absence at the university level (approved) in advance.

Algorithms

Algorithm.

- [webster.com] A procedure for solving a mathematical problem (as of finding the greatest common divisor) in a finite number of steps that frequently involves repetition of an operation.

- [Knuth, TAOCP] An algorithm is a finite, definite, effective procedure, with some input and some output.

> Great algorithms are the poetry of computation. Just like verse, they can be terse, allusive, dense, and even mysterious. But once unlocked, they cast a brilliant new light on some aspect of computing. - *Francis Sullivan*

Definition of Algorithm

must have a well-established structure in terms of the order of their execution, but not necessary in a sequencing. e.g. parallel algorithms, GPU, Interrupt

An algorithm is an **ordered** set of **unambiguous**, **executable** steps that defines a **terminating** process.

The information in the state of the process must be sufficient to determine uniquely and completely the actions required by each step

Make a list of **all** the positive integers

The execution of an algorithm must lead to an end

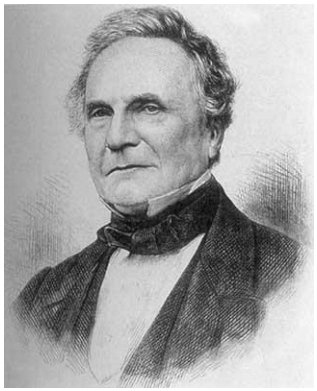
Etymology

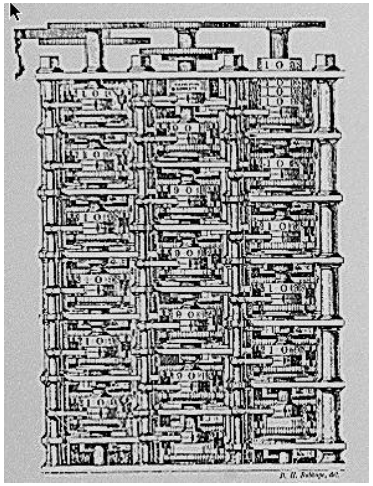
Etymology [ˌetɪˈmaləʤi]. [Knuth, TAOCP]

- *Algorism* = process of doing arithmetic using Arabic numerals.
- A misperception: *algiros* [painful] + *arithmos* [number].
- True origin: Abu 'Abd Allah Muhammad ibn Musa al-Khwarizm was a famous 9th century Persian textbook author who wrote *Kitab al-jabr wa'l-muqabala*, which evolved into today's high school algebra text.

Theory of Algorithms

"As soon as an Analytic Engine exists, it will necessarily guide the future course of the science. Whenever any result is sought by its aid, the question will arise - by what course of calculation can these results be arrived at by the machine in the ***shortest*** *time?* - *Charles Babbage*





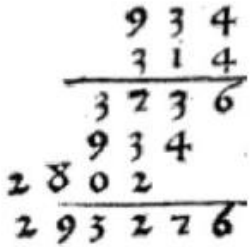
Theory of Algorithms

"As soon as an Analytic Engine exists, it will necessarily guide the future course of the science. Whenever any result is sought by its aid, the question will arise - By what course of calculation can these results be arrived at by the machine in the *shortest time*? - *Charles Babbage*

$$x \cdot y = \begin{cases} 0 & \text{if } x = 0 \\ \lfloor x/2 \rfloor \cdot (y + y) & \text{if } x \text{ is even} \\ \lfloor x/2 \rfloor \cdot (y + y) + y & \text{if } x \text{ is odd} \end{cases}$$

Multiplication by duplation and mediation
In Eastern Europe

Algorithm runs on machines, not manually




```
PeasantMultiply(x, y):
  prod ← 0
  while x > 0
      if x is odd
          prod ← prod + y
      x ← ⌊x/2⌋
      y ← y + y
  return prod
```

x		y		prod
				0
123	:	+456	=	456
61	:	+912	=	1368
30	:	~~1824~~		
15	:	+3648	=	5016
7	:	+7296	=	12312
3	:	+14592	=	26904
1	:	+29184	=	56088

$$x \cdot y = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left(X[i] \cdot Y[j] \cdot 10^{i+j} \right).$$

```
FibonacciMultiply(X[0..m−1], Y[0..n−1]):
  hold ← 0
  for k ← 0 to n + m − 1
      for all i and j such that i + j = k
          hold ← hold + X[i] · Y[j]
      Z[k] ← hold mod 10
      hold ← ⌊hold/10⌋
  return Z[0..m + n − 1]
```

Design and Analysis of Algorithms

The skills required to effectively design and analyze algorithms are entangled with the skills required to effectively describe algorithms. A complete description of any algorithm has four components [Jeff Erickson]:

- What: A precise specification of the problem that the algorithm solves.
- How: A precise description of the algorithm itself.
- Why: A proof that the algorithm solves the problem it is supposed to solve.
- How fast: An analysis of the running time of the algorithm.

Computer programs are concrete representations of algorithms, but algorithms are not programs.

Algorithmic Paradigms

Design and analysis of computer algorithms.

- Greedy.
- Divide-and-conquer.
- Dynamic programming.
- Network flow.
- Randomized algorithms.
- Intractability.
- Coping with intractability.

Critical thinking and problem-solving.

Applications

Wide range of applications.

- Caching.
- Compilers.
- Databases.
- Scheduling.
- Networking.
- Data analysis.
- Signal processing.
- Computer graphics.
- Scientific computing.
- Operations research.
- Artificial intelligence.
- Computational biology.
- . . .

We focus on algorithms and techniques that are useful in practice.