

CMSC 351
Introduction to Algorithms

Fall 2019

Administration

- Webpage
 - ▶ Get homework assignments
 - ▶ Syllabus
 - ▶ Other documents
- Piazza
 - ▶ Ask questions
 - ★ Do **not** post solutions.
 - ★ Do **not** ask if your answer or approach is correct.
 - ▶ Discuss issues
 - ▶ Public versus Private
- ELMS & Gradeserver
 - ▶ Get homework solutions
 - ▶ See grades
- Gradescope
 - ▶ Hand in homework
 - ▶ See graded homeworks and exams

Administration (continued)

- Textbook (bookstore/on reserve at McKeldin Library)
 - ▶ Cormen, Leiserson, Rivest, and Stein, *Introduction to Algorithms* (3rd ed., 2009). MIT Press. (Any edition is fine.)
- Homework
 - ▶ Regular homeworks: typically due each Friday.
 - ▶ Must be in PDF.
 - ▶ Must be easy to read (your responsibility).
 - ▶ Late date: 25% off your actual grade. (One get-out-of-jail-free card.)
 - ▶ **Your neighbor should understand your answers.**
 - ▶ **Study groups.** State who is in your study group at top of homework.
 - ▶ **Must write up homework solutions yourself.**
 - ★ State what outside resources you used to solve each problem.
 - ▶ **Do problems from book (and other books).**

Administration (continued)

- Class attendance
 - ▶ You are responsible for what is said in class.
 - ▶ Laptops and other devices: **Do not share during class.**
 - ▶ Lectures will be posted (mostly).
- Office hours
- Grading
- Exams
 - ▶ Two evening midterms: **6:00-8:00pm.**
 - ★ Thursday, October 3rd
 - ★ Thursday, November 7th
 - ▶ Final exam: **4:00-6:00pm.**
 - ★ Friday, December 13th
- Academic integrity.

Topics (tentative)

- Introduction, Ch. 1,2
- Quadratic sorting algorithms
- Mergesort, Ch. 2
- Summations, Appendix A
- Recurrences, Ch. 4
- Heapsort, Ch. 6
- Quicksort, Ch. 7
- Sorting in Linear Time, Ch. 8
- Medians and Order Statistics, Ch. 9
- Graphs and Trees, Appendix B
- Minimum Spanning Trees, Ch. 23
- Shortest Paths: Dijkstra's algorithm, Ch. 24.3
- Introduction to NP-completeness, Ch. 34

Origin of the word, Algorithm

Origin of the word, Algorithm

- A corruption of the name of 9th century Persian mathematician, Muhammad ibn Musa al-Khwarizmi

Origin of the word, Algorithm

- A corruption of the name of 9th century Persian mathematician, Muhammad ibn Musa al-Khwarizmi
- “Muhammad, son of Moses, the Khwarizian”

Origin of the word, Algorithm

- A corruption of the name of 9th century Persian mathematician, Muhammad ibn Musa al-Khwarizmi
- “Muhammad, son of Moses, the Khwarizian”
- khwarizm is an ancient city, now called Khiva, in the Khorezm province of Uzbekistan.

Origin of the word, Algorithm

- A corruption of the name of 9th century Persian mathematician, Muhammad ibn Musa al-Khwarizmi
- “Muhammad, son of Moses, the Khwarizian”
- khwarizm is an ancient city, now called Khiva, in the Khorezm province of Uzbekistan.
- Al-Khwarizmi is best known for Al-Kitab al-mukhtasar fihisab al-gabr wa'l-muqabla

Origin of the word, Algorithm

- A corruption of the name of 9th century Persian mathematician, Muhammad ibn Musa al-Khwarizmi
- “Muhammad, son of Moses, the Khwarizian”
- khwarizm is an ancient city, now called Khiva, in the Khorezm province of Uzbekistan.
- Al-Khwarizmi is best known for Al-Kitab al-mukhtasar fihisab al-gabr wa'l-muqabla
- from which the modern word algebra derives

Why learn this material?

Why learn this material?

- Algorithms are everywhere in Computer Science (and elsewhere).

Why learn this material?

- Algorithms are everywhere in Computer Science (and elsewhere).
- Useful for later courses.

Why learn this material?

- Algorithms are everywhere in Computer Science (and elsewhere).
- Useful for later courses.
- Useful for computer programming.

Why learn this material?

- Algorithms are everywhere in Computer Science (and elsewhere).
- Useful for later courses.
- Useful for computer programming.
- Useful to get a job.

Why learn this material?

- Algorithms are everywhere in Computer Science (and elsewhere).
- Useful for later courses.
- Useful for computer programming.
- Useful to get a job.
- Useful on the job.

What is an algorithm?

Definition

An *algorithm* is a finite list of step-by-step instructions for solving a problem.

Efficiency

- Time
- Space

Example

Tournament assignment. (Think about at home.)

Runtimes are Critical

Example

Two algorithms:

- Insertion sort: $2n^2$
- Merge sort: $50n \lg n$

Two computers:

- Computer A runs 10 Billion instructions / second
- Computer B runs 10 Million instructions / second

Compute the time to sort 10 Million numbers:

- Computer A uses Insertion Sort
- Computer B uses Merge Sort

Calculate Time

Example

Insertion Sort 10 Million numbers on Computer A:

Calculate Time

Example

Insertion Sort 10 Million numbers on Computer A:

$$\frac{2 \cdot (10^7)^2 \text{ instructions}}{10^{10} \text{ instructions / second}} = 20000 \text{ seconds} \approx 5.5 \text{ hours}$$

Calculate Time

Example

Insertion Sort 10 Million numbers on Computer A:

$$\frac{2 \cdot (10^7)^2 \text{ instructions}}{10^{10} \text{ instructions / second}} = 20000 \text{ seconds} \approx 5.5 \text{ hours}$$

Merge Sort 10 Million numbers on Computer B:

Calculate Time

Example

Insertion Sort 10 Million numbers on Computer A:

$$\frac{2 \cdot (10^7)^2 \text{ instructions}}{10^{10} \text{ instructions / second}} = 20000 \text{ seconds} \approx 5.5 \text{ hours}$$

Merge Sort 10 Million numbers on Computer B:

$$\frac{50 \cdot 10^7 \lg(10^7) \text{ instructions}}{10^7 \text{ instructions / second}} = 1163 \text{ seconds} \approx 20 \text{ minutes}$$