# Welcome to Analysis of Algorithms (CS583 - 004)

Amarda Shehu

Spring 2019

Amarda Shehu Welcome to Analysis of Algorithms (CS583 - 004)

### **Class Information**

#### Tentative Syllabus

#### Information

#### Instructor: Amarda Shehu amarda\AT\gmu.edu

Place and Time: Innovation Hall 203, F 1:50-4:10 pm Instructor Office Hours: ENGR #4452, F 11:30 am - 1:30 pm TA Office Hours: Yang Yong (yyang29\AT\gmu.edu), ENGR #4456, W 4:00 - 6:00pm

#### Tentative Syllabus

Date	Topic	Chapters	Assignments
Jan25	Course Overview, Insertion Sort, Merge Sort	C1-3, [pdf], [pdf]	Self-eval. Quiz
Feb01	Asymptotic Notations, Bounding Functions	C3-5, [pdf]	Quiz

#### Sorting and Order Statistics

Feb08	More on Bounding Functions, Bounding Recurrences	C3-5, [ <u>pdf]</u>	Quiz, Hw1 Out
Feb15	More on Bounding Recurrences	C3-5, [pdf]	Quiz
Feb22	Average Time Analysis and Quicksort	C6-8, [pdf]	Quiz
Mar01	Heapsort, Comparison- vs. Linear-time Sorting	C6-8, [pdf]	Quiz, Hw1 Due
Mar08	Order Statistics	C9, [pdf]	Quiz
Mar22	Exam 1		Exam 1

#### Data Structures for Storing and Searching

Mar29	Hashing	C11 [pdf]	Quiz
Apr05	Advanced Data Structures and Analysis	C12, C13 [pdf]	Quiz

#### Optimization and Graph Algorithms

Apr12	Dynamic Programming, Greedy Algorithms	C15-16, [pdf]	Hw2 Out, Quiz
Apr19	Graph Representation, Uninformed Search, Applications	C22, [pdf], [pdf]	Quiz
Apr26	Informed Graph Search	C24-25, [pdf], [pdf]	Quiz
May03	MST, Maximum Flow	C23, C26 [pdf], [pdf]	Quiz, Hw2 Due
May10	Exam 2	Innovation Hall 203	1:30 pm 4:15 pm

#### Instructor: Amarda Shehu

Office: ENG #4452 Email: amarda AT gmu.edu Web: cs.gmu.edu/~ashehu

#### CS583 Hours

Class: F 1:30 - 4:10 pm Place: Innovation Hall 203 Office Hours: F 11:30 am - 1:30 pm

#### TA:

#### Email: yyang29 AT gmu.edu ENG#5321, W 4:00 - 6:00 pm



- 2 Outline of Today's Class
- The Importance of Designing and Analyzing Algorithms
  The Pervasiveness of Algorithms in Our Society
  What does It Take to Design Useful Algorithms?

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

# Why are we Here?

- In Calculation with Hindu Numerals, 825 A.D., Muhammad ibn Musa al-Khwarizmi introduced Indian decimal system
- The book was translated into latin in 12<sup>th</sup> century as *Algoritmi de numero Indorum*
- algorithm was introduced to refer to a procedure for calculations with numbers
- Short answer: We are here to design and analyze algorithms - procedures to solve useful problems



Figure: Soviet stamp for al-Khwarizmi's 1200<sup>th</sup> birthday. ©wikipedia.

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

### Seach Engines: Google, Yahoo, Ask



Figure: Searched for *algorithms for life sciences* in text in the web.

- Pattern matching algorithms and information searching algorithms are fundamental to our ability to parse through an overwhelming amount of information
- Google was founded on the ability of two Stanford University Ph.D. students, Sergey Brin and Larry Page, to design a fast information searching algorithm, *BackRub*.

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

# Orientation Software: Google maps, GPS navigators

🗶 a secondario de presenta de la construcción de l	10
🖕 🖞 v 🔥 🖓 📚 🚺 Hits Image consistence This and takenal	-
Manufacture Control Control and Andrews	~
Pres veces + Consid States - Construction	
Anada shere, having 0 0 Mataminal on Hala a che. 0 3 Hor On High to Google Hal. 0	-
Google Ready Maps. She asard attern and places of inferred. (and report	
Get Directions McMaps	<u>unk</u>
IDE Mark H. Hueldon, IT 77000 (No. 1996)     IDE Vander H. TYTON (No. 1996)     I	
Virtual directions to George Masso University-pearlise, 4400 University Dr. 1 2017 or 1 de 21 lanes (Dense Virtual directions) 2 1400 Martines III 7 1000	No. of the local division of the local divis
L Head methods to Hale Stream 1 1 11	
2. At the traffic ords, take the Brill exit. (B) 3.4 million and an annual second seco	
3. Turn right at Southenare Died 🛞 12 ml Raw Streem	
A Durn left at Sam jacinto St. B 1.4 million annual Anamata Statute annual Charter annual Charter annual Charter annual Charter	
5. Tues sight to merge onto US-59 N B 2.8 m and have have been been been been	
C. Take the set onto 1-10 E feward     B 273 m     Description     Descri	
7. Take and 1590 on the left to marge B 15.7 m	
Sale est 458 to merge onto 169 N III 5.3 million and the sale of the sale	
Castland of MARY     Finite Street And	
20. Take est 106 to merge onto 1-66 N 20.4 mi	
11. Take est 2001 50 marge ords 140 N 122 m Insel Godden Entrong Derga	
12: Take the exit onto IQ21 Externel Chattanacoga Entroy of Externel control (2) = 1 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 -	
33. Tale and 1098 to recept or to 179 H (8) 100 million (1999) and the second second sectors are to second sectors (1999) and the second se	

# Figure: Output of a path finding algorithm.

- Path from Rice University, Houston, Texas to George Mason University, Fairfax, Virginia
- Path finding algorithms can be found in portable GPS navigators
- Most versions of the algorithm work with a static map (static conditions on the ground)

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

#### Exploration, Search and Rescue, and Motion Planning



Figure: Ron Li and his research team are developing algorithms to help the rovers, Spirit and Opportunity, to navigate and find a safe path to a winter resting area. ©NASA.



Figure: Erion Plaku at Rice University is developing algorithms that plan paths for car-like robots in cluttered environments. ©E. Plaku.

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

### Simulating Molecular Properties for Drug Design



Figure: Successful docking of HIV protease with a small inhibitor ligand. ©A. R. Leach.



Figure: Simulating the ability of proteins like ubiquitin to change shape as needed to accommodate and dock with different partner molecules. ©A. Shehu.

### What is an Algorithm?

 Recipe, computational procedure that transforms input into output, tool to solve well-defined problems, sequence of instructions



• An algorithm is a finite sequence of unambiguous instructions for completing a task, that:

### What is an Algorithm?

 Recipe, computational procedure that transforms input into output, tool to solve well-defined problems, sequence of instructions



- An algorithm is a finite sequence of unambiguous instructions for completing a task, that:
  - when given an initial state ( $\geq 0$  inputs),
  - proceeds through a well-defined series of successive states,
  - eventually terminating in an end-state ( $\geq$  outputs)

## What is an Algorithm?

 Recipe, computational procedure that transforms input into output, tool to solve well-defined problems, sequence of instructions



- An algorithm is a finite sequence of unambiguous instructions for completing a task, that:
  - when given an initial state ( $\geq$  0 inputs),
  - proceeds through a well-defined series of successive states,
  - eventually terminating in an end-state ( $\geq$  outputs)
- Examples of algorithms?

# What is an Algorithm?

• Recipe, computational procedure that transforms input into output, tool to solve well-defined problems, sequence of instructions



- An algorithm is a finite sequence of unambiguous instructions for completing a task, that:
  - when given an initial state ( $\geq$  0 inputs),
  - proceeds through a well-defined series of successive states,
  - eventually terminating in an end-state ( $\geq$  outputs)
- Examples of algorithms?

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

Why do we Write Algorithms?

- Model it as a well-defined computational problem
- Is the solution to the problem *computable*?

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

### Why do we Write Algorithms?

- Model it as a well-defined computational problem
- Is the solution to the problem *computable*?
- If it is, does there exist an *efficient* algorithm for it?

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

### Why do we Write Algorithms?

- Model it as a well-defined computational problem
- Is the solution to the problem *computable*?
- If it is, does there exist an efficient algorithm for it?
- How do we come up with an efficient algorithm?

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

### Why do we Write Algorithms?

- Model it as a well-defined computational problem
- Is the solution to the problem *computable*?
- If it is, does there exist an efficient algorithm for it?
- How do we come up with an efficient algorithm?

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

#### How do we Write Algorithms?

#### • Are there any useful paradigms from which we can choose?

• How do we know the algorithm we have written is correct?

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

- Are there any useful paradigms from which we can choose?
- How do we know the algorithm we have written is correct?
- Can we improve the algorithm? On what do we improve?

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

- Are there any useful paradigms from which we can choose?
- How do we know the algorithm we have written is correct?
- Can we improve the algorithm? On what do we improve?
  Efficiency?

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

- Are there any useful paradigms from which we can choose?
- How do we know the algorithm we have written is correct?
- Can we improve the algorithm? On what do we improve?
  - Efficiency?
    - Less space? Storage Concerns?
    - Shorter running time? Faster is always better.

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

- Are there any useful paradigms from which we can choose?
- How do we know the algorithm we have written is correct?
- Can we improve the algorithm? On what do we improve?
  - Efficiency?
    - Less space? Storage Concerns?
    - Shorter running time? Faster is always better.
  - Can we make the algorithm simpler?

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

- Are there any useful paradigms from which we can choose?
- How do we know the algorithm we have written is correct?
- Can we improve the algorithm? On what do we improve?
  - Efficiency?
    - Less space? Storage Concerns?
    - Shorter running time? Faster is always better.
  - Can we make the algorithm simpler?
    - Use a simpler data structure?
    - Allow developers to extend and generalize?
    - Aesthetics concerns?

The Pervasiveness of Algorithms in Our Society What does It Take to Design Useful Algorithms?

- Are there any useful paradigms from which we can choose?
- How do we know the algorithm we have written is correct?
- Can we improve the algorithm? On what do we improve?
  - Efficiency?
    - Less space? Storage Concerns?
    - Shorter running time? Faster is always better.
  - Can we make the algorithm simpler?
    - Use a simpler data structure?
    - Allow developers to extend and generalize?
    - Aesthetics concerns?

# Paradigms we Will See in this Class

- Brute force
- Divide and conquer
- Decrease and conquer
- Transform and conquer
- Space and time tradeoffs
- Dynamic Programming
- Greedy Approach
- Iterative improvement
- Backtracking
- Branch and bound