# *CS202 - Algorithm Analysis*
## **An Introduction**

Aravind Mohan

Allegheny College

January 16, 2023

- Lecture Session:
  - Tuesday and Thursday
    8:00 AM - 9:15 AM, (Alden-109)
- Lab Session:
  - Friday 2:30 PM - 4:20 PM, (Alden-109)

# Professor's Office Hours

- Monday, Tuesday, Wednesday, and Thursday: 10:00 AM - 12:00 PM
- Also by appointment

**Discord/Email to schedule time outside office hours.**

To schedule an office hours time slot, please visit my website [teaching page] and click on the
**Schedule Meeting** link located on the top right-hand corner to schedule 20 mins slots.

**Let us connect with each other and enjoy our time together...**

# Website Details

- Professor's Website:

  `https://www.cs.allegheny.edu/sites/amohan/`

- Course Website:

  `https://www.cs.allegheny.edu/sites/amohan/course.php?cid=OQ==`

# Administrative Stuff!

- Please verify if you are correctly registered for the course using Self Service.
- Make sure to read through the Syllabus before next class.

| Lab Assignments | 35% |
| Quizzes | 10% |
| Midterm Exam | 10% |
| Final Exam | 15% |
| Course Project | 20% |
| Class Participation | 10% |
| | |

**Please read the Syllabus to get an overview of the course.**

- Any question is a valid question. There is no question which is good and bad. So, questions are always welcome.
- Interaction is the best way to get rid of long lectures. So, let us try to interact more so that the communication is a two way process and the class is not boring.
- Reach out to me to ask for any Accommodations ...

**We will work together to retain Algorithmic Knowledge from this course.**

- Attentively listen to classes and try to participate in all class discussions.
- Take detailed notes during every class period.
- Clarify with the Professor, if a lesson is confusing.
- Complete all the reading assignments thoroughly.
- Do the in-class exercises thoroughly.

**Be ready to think, process, and learn visually in this course!**

- An algorithm is a finite set of instructions that if followed, accomplishes a particular task.
- Algorithm is written to make something **fast**, and/or **solve some interesting problems**.
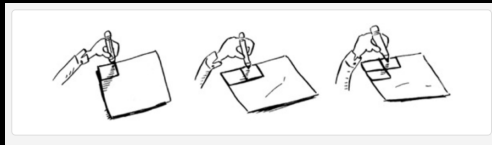- The word Algorithm comes from the name of a Persian author:

**Abu Jafar Mohammed ibn Musa al Khowarizmi.**
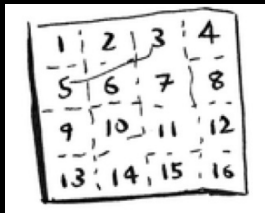
- Draw a grid of 16 boxes as shown in the figure above.
- Come up with an Algorithm to solve this problem?

- **Algorithm 1:** One way to do this is to draw 16 boxes, one at a time.
- It takes **16** steps to complete this task.

Can we solve this problem faster in **4** steps?  **How?**

**Hint: Try solve the problem without using Pen/Pencil ...**

- Provide your answers in the class activity sheet.

- Leon works at the aquarium. When he tries put each turtle in its own tank, he has one turtle too many. But if he puts two turtles per tank he has one tank too many.

  How many turtles and tanks does Leon have? Assume maximum number of tanks in the aquarium to be 10.

- Come up with an Algorithm to solve this problem? Now you may approach this trial and error first, to identify the math to include in the algorithm.

- Complete **aquarium.py** to implement the algorithm and provide your completed code in the class activity sheet.
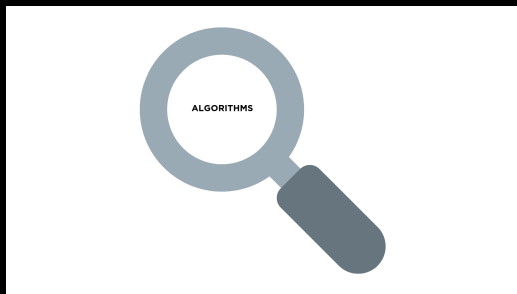
Sorting Algorithms

- **Sorting:** For example, sorting music playlist data and rank the songs based on user's interest.
- For example, sorting patient data and rank the patients based on their severity level.
- and more · · ·

- **Searching:** For example, searching a phone book and finding the phone number associated with a person.
- Searching for a patient's history and finding if the patient had any allergies to medications.
- and more $\cdots$

# What Algorithmic Problems do we learn?



- **Graph:** For example, finding the shortest distance between two locations (similar to GPS!)
- Recommend friends based on a person's connections in a friend's network. (similar to Facebook!)
- and more · · ·

## TEXT

- **String Algorithms:** For example, finding longest common subsequence in Strings.
- and more $\cdots$

**By end of this course, you'll in general**

- Master a variety of algorithms.
- Be well equipped to learn advanced algorithms in the field of AI, Databases, Cloud Computing, and so on ...
- Be prepared to take on bigger challenges on your senior thesis and at work after graduation.

## What do we do in Labs?

- Combination of individual and team-based labs.
- Solve algorithmic problems in a non-programmatic manner.
- Develop algorithms using **Python** as the primary programming language.
- Compare the performance of algorithms using Charts.

# A More Formal Example



| S | | | | | | | |
|---|---|---|---|---|---|---|---|
| 156 | 141 | 35 | 94 | 88 | 61 | 111 | 77 |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |

**Algorithm -** Find Most Played Song $(S, n)$
**Input -** A set of play counts associated with a variety of songs inside a playlist.
**Output -** The most played song.

1: $temp \leftarrow S[0]$
2: $res \leftarrow 1$
3: **for** $i = 1$ to $n$ **do**
4:     **if** $S[i] > temp$ **then**
5:         $temp \leftarrow S[i]$
6:         $res \leftarrow i + 1$
7:     **end if**
8: **end for**
9: **return** res;

**Algorithm -** Find Least Played Song $(S, n)$

Provide your answers in the class activity sheet.

**We will try a few variations during the Lab Session this week!**

- Sign up for Course Discord channel. (Link accessible in the course webpage!)
- Read the Syllabus before next class and bring questions to next class.