

Algorithms
(and a little more logistics)

Syllabus

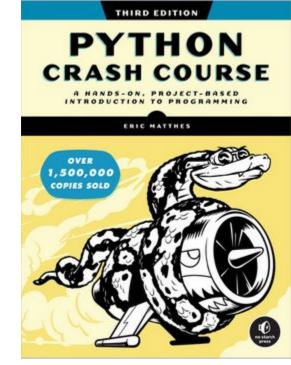
ALL logistics information can be found in the syllabus:

Section 30: Link

Section 40: Link

Textbook

GV lets you access the book for *FREE*

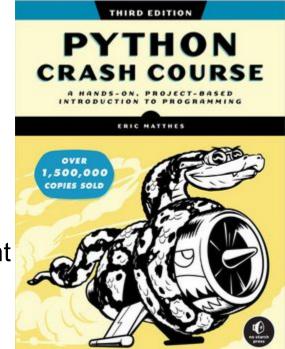


Textbook

GV lets you access the book for *FREE*

Click here to access!

Institution access -> Select GV -> Log into GV account



Textbook

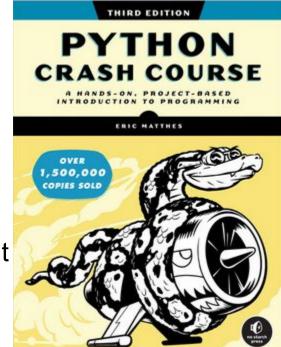
GV lets you access the book for *FREE*

Click here to access!

Institution access -> Select GV -> Log into GV account

Syllabus contains recommended reading

Best done before we cover it in class



Grading

Grading

Assignment breakdown:

Programming Projects (1 – 3)	6%, 6%, 6%
Lab Assignments	18%
Midterms (1 and 2)	13%, 13%
Lab Exam	13%
Final Exam	15%
In Class Activities	10%

Grading

Assignment breakdown:

Programming Projects (1 – 3)	6%, 6%, 6%
Lab Assignments	18%
Midterms (1 and 2)	13%, 13%
Lab Exam	13%
Final Exam	15%
In Class Activities	10%

Grade scale:

A	93%	C	73%
A-	90%	C-	70%
B +	87%	D+	67%
В	83%	D	60%
В-	80%	F	Below 60%
C +	77%		

Remember:

>= 60% average on exams to pass!

Lecture attendance tracked via in-class activities

Lecture attendance tracked via in-class activities

- One "freebie" missed activity
- Two if you go to the Computing Success Center regularly*
- If something drastic happens, let me know!

Lecture attendance tracked via in-class activities

- One "freebie" missed activity
- Two if you go to the Computing Success Center regularly*
- If something drastic happens, let me know!

Lab attendance* is 50% of your lab grade!

Lecture attendance tracked via in-class activities

- One "freebie" missed activity
- Two if you go to the Computing Success Center regularly*
- If something drastic happens, let me know!

Lab attendance* is 50% of your lab grade!

- You really need to be there!
- If something super drastic happens, let me know!

Late days

- Projects will have due dates posted
- Labs should be finished before you leave
 - If not, lab is due 11:59pm the night before the next lab
- Labs and projects can be turned in up to three days late
 - 10% off per day
 - Weekends <u>do</u> count, sorry!
- Again, if life happens, please let me know ASAP!

Academic honesty

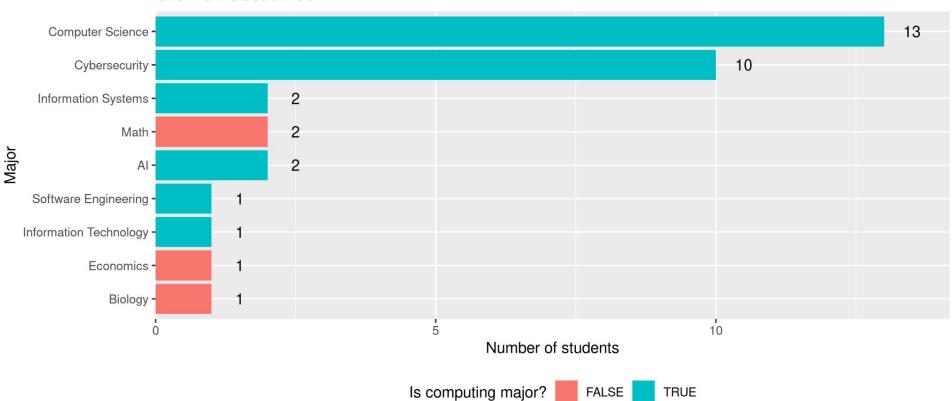
- Document collaboration, internet resources, and Al usage
- You should be able to explain and defend your submission

Helpful links!

- Computing Success Center
- Other learning centers on campus!
- One-on-one tutoring is available
- General GV support page find what you need!
- Counseling Center

Questions about logistics?

CIS 162 Section 30



Section 30, Q2: What do you want out of this course?

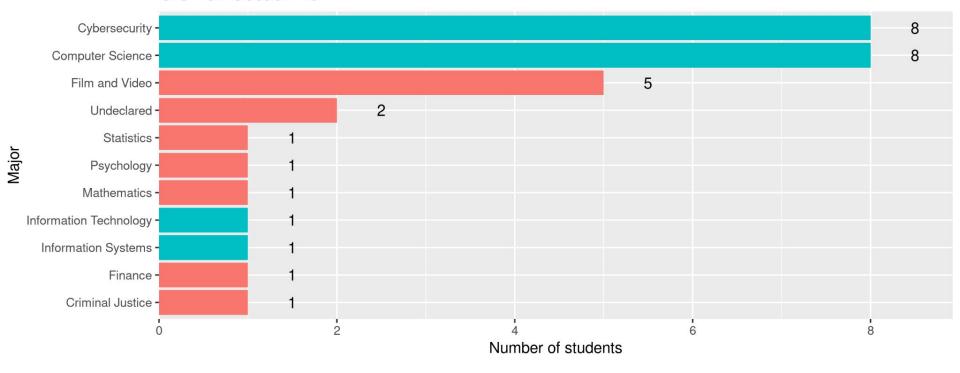
- "Learn more about coding, also to learn how I can use it regularly"
- "A good grade"
- "Learning game dev"
- "A knowledge of where, specifically, I want to take my career in comp. sci."
- "To see if computer science is for me"

 "If I should move forward with my CS major"

Section 30, Q3: Something you're looking forward to

- "I'm excited for the brewing class I am taking"
- "Meeting new people, new adventures, new experiences"
- "I'm excited to go to the football games"
- "Learning things I'm interested in"
- "Ice cream"
- "Christmas break"
- "Hollow Knight Silksong"
- "Excited to meet new people, and be out of highschool"
- "I'm excited to learn"
- "Some of my classes look interesting"

CIS 162 Section 40



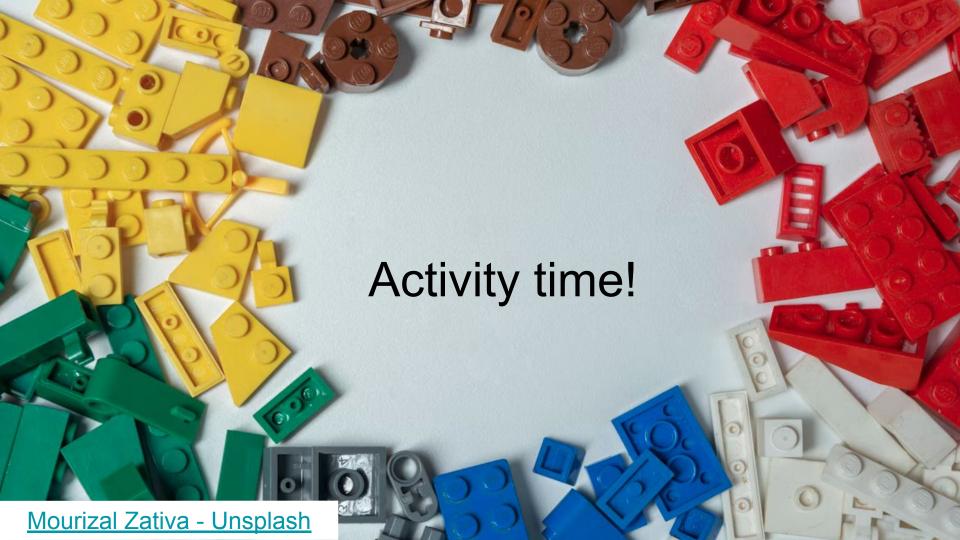
Is computing major? FALSE TRUE

Section 40, Q2: What do you want out of this course?

- "How to code games"
- "My future job"
- "I want to be able to create my own code and actually understand what it all does"
- "I want knowledge to excel in my major, Cybersecurity"
- "I want to gain confidence in the material before I go onto the next course"
- "More comp sci knowledge, maybe a few friends"
- "I want the skills to get a good future job"

Section 40, Q3: Something you're looking forward to

- "I am excited for everything this semester"
- "For it to end"
- "I'm excited to learn, make friends, and just understand more about topics I'm interested in"
- "New food. New apartment. GROWTH"
- "Being on my own, activities on campus"
- "To learn Python"
- "Excited to get into my animation class"
- "Being at university instead of community college"
- "My birthday"
- "Snow"
- "Auditioning for improv club"





What is an algorithm?

What is an algorithm?

Dr. Ferg's answer:

"A set of precise steps to accomplish a specific purpose"

What is an algorithm?

Dr. Ferg's answer:

"A set of precise steps to accomplish a specific purpose"

Formal answer (from CLRS algorithms book):

"Any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output. An algorithm is thus a sequence of computational steps that transform the input into the output."

Outside computers:

Outside computers:

- Recipes
- IKEA furniture instructions
- Driving directions

Outside computers:

In computers:

- Recipes
- IKEA furniture instructions
- Driving directions

Outside computers:

- Recipes
- IKEA furniture instructions
- Driving directions

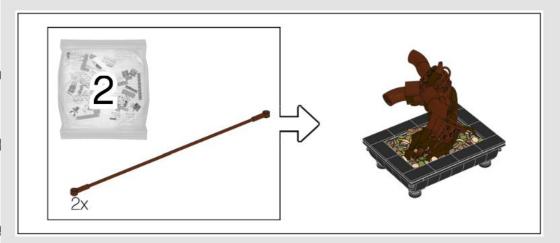
In computers:

- Sorting algorithms
- Pathfinding algorithms
 - E.g., Google Maps
- Cryptographic algorithms
 - E.g., hashing
- Evolutionary search algorithms
- And MANY more!

Examp

Outside (

- Reci
- IKEA
- Drivi











h algorithms

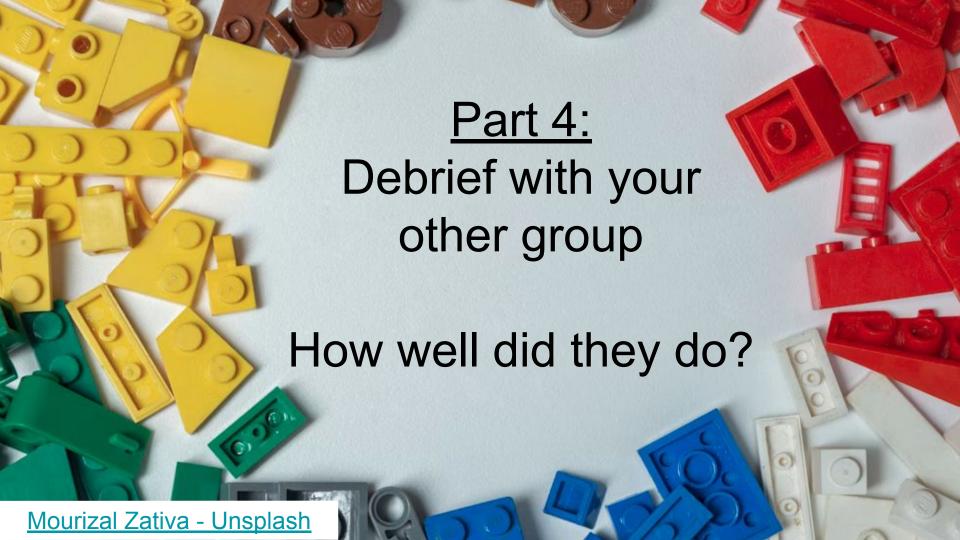
nms laps orithms

22









Class debrief

How did it go? Any challenges?

Class debrief

How did it go? Any challenges?

Which was harder: creating the algorithm, or following it?

Class debrief

How did it go? Any challenges?

Which was harder: creating the algorithm, or following it?

What would you change?

What's the point?

What's the point?

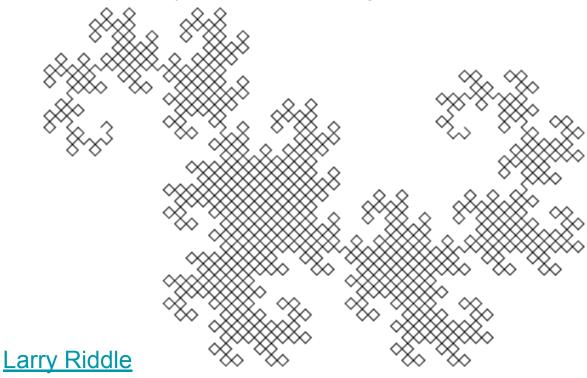
Often, the hard part is coming up with the algorithm

With an algorithm in hand, coding it is often fairly simple

What infrastructure would you need to code a robot to build your LEGO?

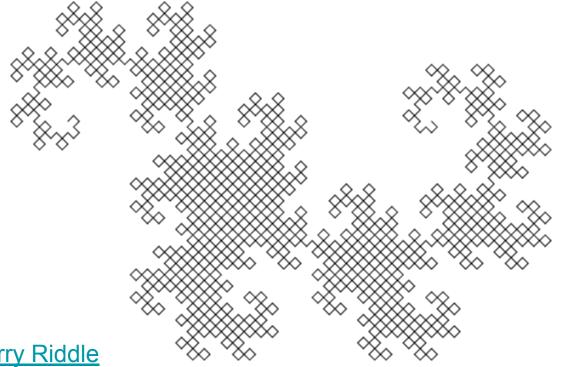
Another example

How would you create an algorithm to make a "Heighway dragon curve"?



Another example

How would you create an algorithm to make a "Heighway dragon curve"?



How about now?

Larry Riddle

Another example

How would you create an algorithm to make a "Heighway dragon curve"?



Applying to CS

My recommended approach when coding:

- Sketch out problem and try to create an algorithm on paper
 a. Working through examples can be great!
- 2. Code a solution from the paper

Many people go straight to code and get bogged down! Focus on one problem (algorithm vs code) at a time!!!