GW

School of Engineering & Applied Science

BE A PART OF IT

THE GEORGE WASHINGTON UNIVERSITY

Computer Science at GW

Prof Bhagi Narahari

what do Computer Scientists do….

What is Computer Science

- Use Computational thinking (& tools) to solve problems in
  - Engineering, Medicine, Science, Law,
  - Arts, Entertainment, Business, Finance…
  - Pretty much everything….
- Computational techniques as third pillar in today's scientific methods
  - Experiments, Theory, Simulation/Computing
- You've been learning about computational thinking via your lab experiments & Python

MythBusters

- CS is just Programming…?
- Software developers sit at their cubicle and talk to their computer all day ?
- The field of CS has no impact on society ?
  - i.e., there is no such thing as Computer Science for Social Good ?

Reality:

- Programming is just one of many tools to solve a problem
  - If CS is about programming then Astronomy must be about telescopes ?!
- Computer Science is all about problem solving and then getting the computer to implement your solution!
- Every software engineer has to work in teams and learn how to communicate their ideas…more so if they are tackling a "real" problem
  - As early as sophomore year, courses require teamwork
- CS is driving social change in today's world
  - Our faculty (CS@GW) leading the way in 'Computing for Social Good'

Solving Problems & Computational Thinking – What is involved?

- Problem solving using computers…process:
  - Design a solution – model the problem & develop the algorithm
  - Design software to implement the solution
    - Programming
- How do we measure how “good” the solution is ?
  - Think like an engineer !
  - All about efficiency – minimize steps needed to complete the task
    - Time to solve the problem, Cost, Ease of deployment,…

Question….your first "algorithm"

- I thought of a number between 1 and 32 – guess what it is by asking me a question
- Smaller the number of guesses the better your solution
  - You can ONLY ask a question of the form:
  - "is it more than X" or "is it less than or equal to X" or "is it equal to X"
- Count the number of questions: what is the maximum number of questions you need to ask to determine the number ?
  - Think of a strategy that minimizes the number of questions
  - Can you generalize your answer/strategy to guess a number between 1 and N ?
- Why is the number of steps/guesses important ?
  - Measures the amount of work - i.e., steps needed to solve the problem = time taken by the program

An "efficient" solution

- Is the number less than or equal to another number X
  - Start: is it less than or equal to 16
    - If answer is yes, then next question is "is it < = 8"
  - At each step you halve the range of numbers you are searching
- Worst case (maximum number of steps)
  - 1. " <= 16 ?"
  - 2. "<= 8?"
  - 3. "<= 4 ?"
  - 4. "< = 2 ?"
  - 5. "< = 1 ?"

So how "good" was your solution- Binary Search: let's do some math!

- For N numbers
- How many numbers are you "searching" at each step:
  - first N, then N/2 and then N/4,… and finally 1 =$N/2^K$
  - Solving for K….
- Number of steps: **$K = \log_2 N$**
  - If N=100, then (log N)= 7 steps
  - If N=1,000,000 then (log N)= 20
- What N are we talking about…..
  - Facebook: scans over 100 terabytes per day! And over 100 petebytes (2^50) !!!
- Next step: Programming the solution!

Why is efficient search important? How did this change our lives ?

CS Program @ GW

- Degrees are BS or BA
  - BA requires double majors or multiple minors
  - BS offers flexible specialization in CS
- option to specialize in a technical area after core competence
  - Cyber Security specialization, Data science, ….
- Flexibility to take a lot of non-CS courses
  - business, economics, criminal justice, Int. Affairs
  - Programs with Business school, Corcoran, Public health,..
  - Easy/flexible path to pursue double majors (lot of CS students do double majors or minors)
  - Some cool project based “student led courses” – interview prep, Rasp Pi dev?
- Research with faculty
- Lots of internships
- Study-abroad-semester built into curriculum

CS Curriculum – First two years

- Year 1: baby steps
  - Intro problem solving using computers: programming, data structures, algorithms
  - Math & Science, includes Discrete math for CS (Analog vs Digital world)
- Year 2: foundational stuff
  - Software engineering & Database systems: building & using S/W
  - Computer Architecture & Systems Programming (hardware & systems)
  - More Math for CS
- You will know enough CS after Year 2 to go into technical depth
  - Sophomores have interned at FAAMG
    (Facebook, Amazon, Apple, Microsoft, Google)…..
  - Sophomores have done research (published papers)
- Comment: CS@GW curriculum is a bit front loaded – we teach you all the foundations by semester 5

CS Curriculum – Years 3,4

- 5th Semester: Fun begins!...
  - Algorithms – all about problem solving
  - Operating Systems – all you want to know about how systems are built
  - After this semester, you are on your way!
- 6th semester: ALL electives…time to PARTY ??
  - Study abroad..and take whatever you want
  - Start technical electives (specialization?) or second major/minor or research
- Senior Year:
  - Year long capstone: design and build your project…entrepreneurship, presentation skills,etc.
  - All your CS courses are electives – focus on what you are interested in!

What are Technical Electives/Tracks..

- Areas synergistic with faculty research
- Graphics and Animation
- Data Science/Analytics – natural language processing
- Artificial intelligence, Machine Learning
- Software engineering and systems
- Computer Security…..

Life after CS@GW

- There are TONS of jobs in CS
  - Startups, Fortune500, Defence, Labs, Govt……
- Where do our graduates go:
  - Industry: Disney, Apple, Cisco, Google, Microsoft, Twitter, Buzzfeed, Facebook
  - Startups…
  - Govt: NASA, Naval Res.Labs, NIH,…
  - Defense: Lockheed-Martin, Raytheon, …
  - Grad schools: Stanford, Penn, Cornell, MIT, Berkeley, CMU, Columbia, Princeton, Georgia Tech, UT-Austin, ….GW (5-year BS+MS or BA+MS)

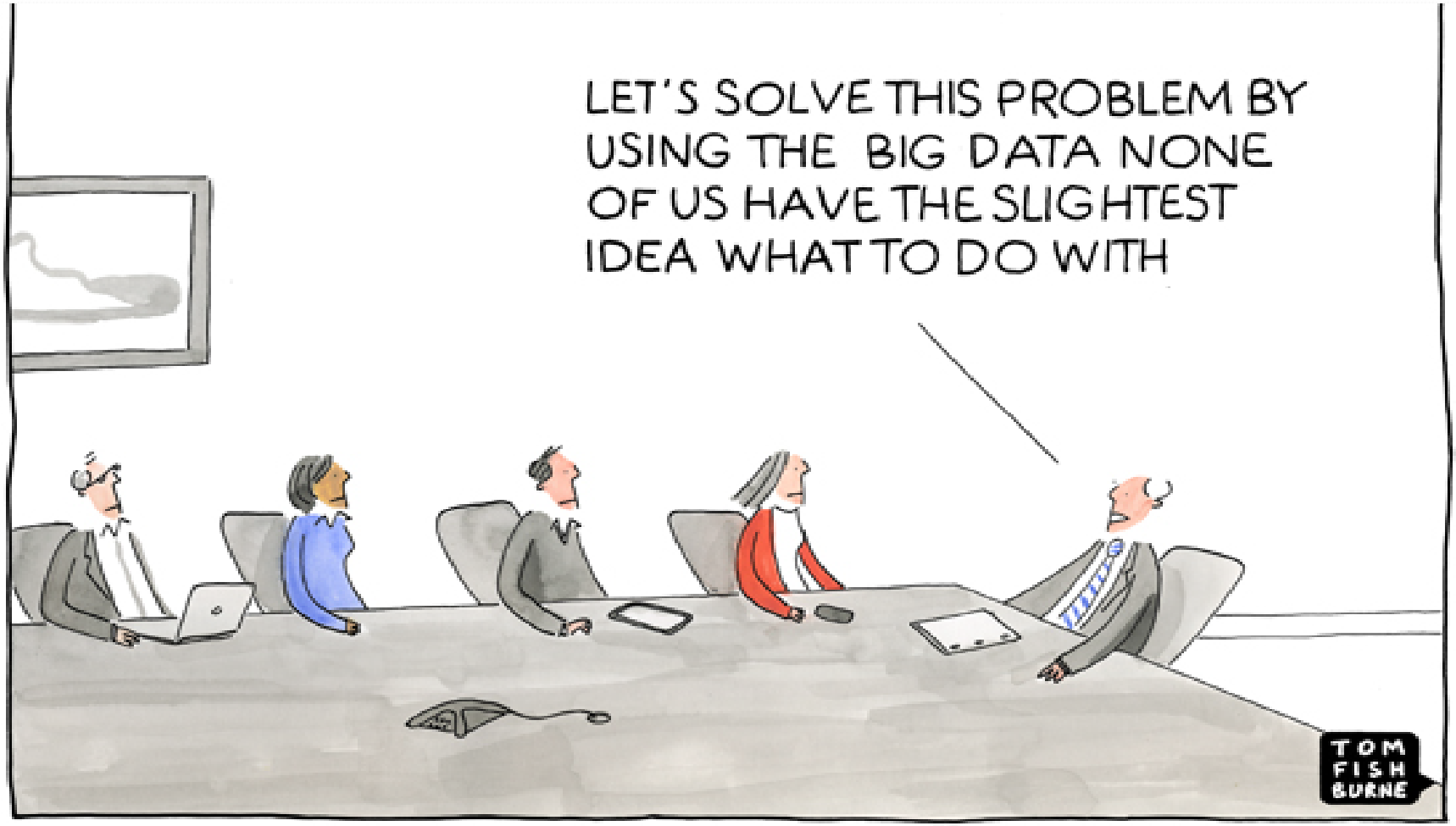
What are some of the topics and areas that we teach and/or research at CS@GW

Our theme: **CS and Social Impact**

- AI for good
- Privacy & Secure elections
- Making mobile devices safer
- Medical computing – robotic surgery
- Institute for Data, Democracy & Privacy

- Many undergraduates work/research with our faculty

Big Data/Data-Science anyone ?

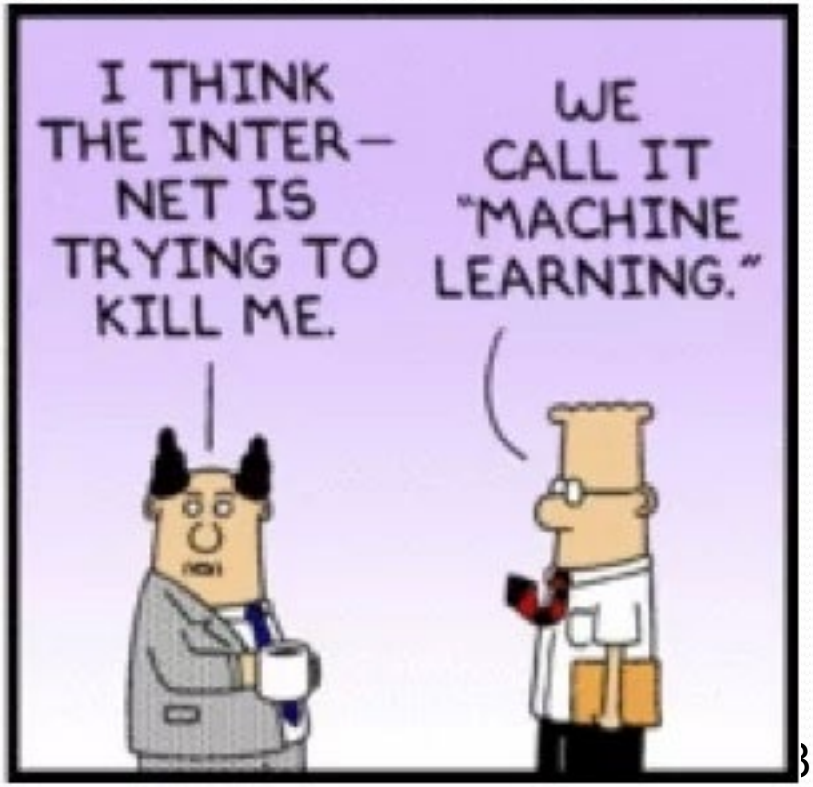


Machine Learning & Artificial Intelligence

Dr Arora

Dr Kinga

Robert Pless, Chair



Natural Language Processing: Why is it hard

- Can Siri understand any question you ask ?

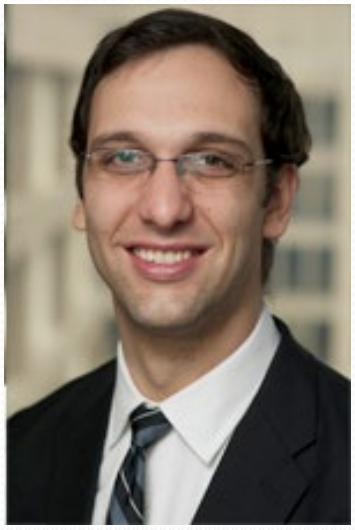
- What does this sentence mean:
- “I made her duck”

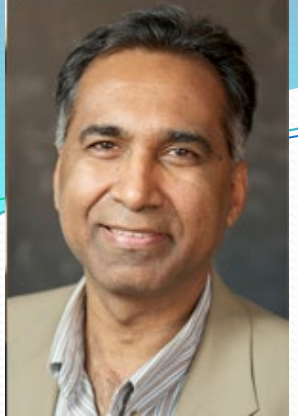
AI for Social Good

- **Bias in AI**: Uncover and quantify human-like biases in machines
  - Discrimination by ride companies (Uber&Lyft), Autonomous cars, Bail approval software, Google Translate
  - Social networks – disinformation analysis and prediction
- Recent result: Uber & Lift charged more if pickup/destination had a higher percentage of (a) non-white residents, or (b) low-income residents or (c) education
  - Has led to a law-suit against Uber
- **Computer vision:** Prof Pless lab
  - Social computing from visual perspective
  - Social media analytics – Eating disorders
  - Creating Image analysis tools to fight sex trafficking
    - FBI and DOJ

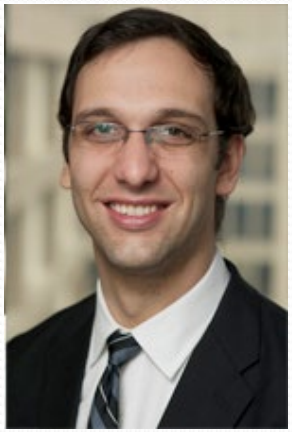
Software Systems: Cloud Computing & Operating Systems

- Efficient management of the system resources
  - Produce results in timely manner – real-time computing
  - Manage cloud computing resources – failure, speed,etc.
- Analogy: Systems SW is like working on the car engine
- Prof Gabe Parmer Prof. Tim Wood
  - Lots & lots of undergraduates (10-20) work in the Systems Lab





Simha



Parmer

Choi

IoT (Internet of Things) & Embedded Systems

Rasp.Pi = IoT platform ?

Cool 1 credit projects course on Rasp.Pi IoT Apps

Graphics & Animation

Prof James Hahn
Virtual Reality & Medicine

CyberSecurity and Privacy

Aviv

Vora

Arkady

Acar

CyberSecurity @ GW

- Usable Security & Privacy – Human factors in design of secure systems including mobile phones: Profs Aviv, Acar

- Cryptography for Big data, Privacy, and fighting fake news – Prof Arkady Yerkumovich
- Secure voting and Privacy – Vora

All of these research groups have undergraduates

New York Times
NPR Morning Edition
WAMU News (local NPR affiliate)
Wired
Computer World

IEEE Spectrum
SIAM News
CNet News
C-Span
msnbc.com
Voice of America

Protecting our Democracy: Secure & Verifiable Voting

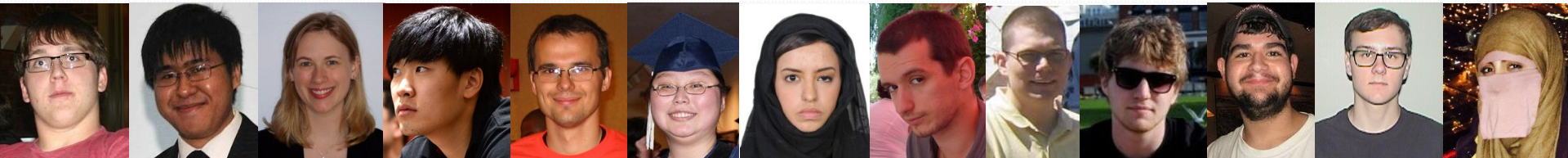
- ***Statistical election audits: 2020: GA? PA?***
- Cryptographic ***voter-verifiable*** voting systems
  - Used by **City of Takoma Park** for city elections
  - **2009, 2011**
- Undergraduates play a key role in all sponsored research











Questions ?

Today's Exercise: security and privacy and building an encryption module!

- First understand the problem and design a solution
- Do you care about privacy ?
- Next implement an "application" in Python
  - Application: You want your photograph(s) to be seen only by authorized people

Security& Privacy Exercise: Encryption

- Encryption – coding your message
  - Sending secrets
  - Safeguard your private information!
- Caeser's Cipher – a simple 'substitution cipher' algorithm
  - History: used by Julius Caeser to send military secrets
- Original Form: Shift each alphabet by 3
  - A replaced by D, B replaced by E,…..Y replaced by B
  - Circular shift
  - "FRIDAY" encrypted as " IULGDB "

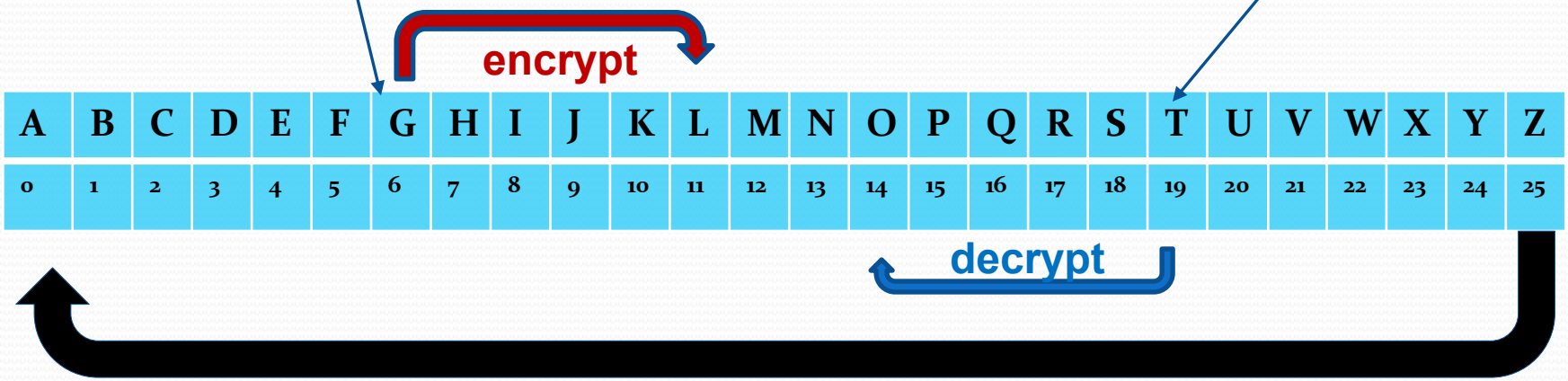
Generalized Shift(Caeser) Cipher.

- Instead of shifting by 3, shift by some secret value K
  - K is between 0 and 25
    - Why ? Because there are 26 letters in the alphabet
- The value K is your secret “Key” (like a password)
- Encryption “algorithm” : Shift each letter (right) by K
- To “decrypt” the message: Shift ‘left’ each letter by K
- Some math: we can assign a number from 0 to 25 to each letter in the alphabet starting with A
  - Shifting by K means adding K to that number
  - But circular addition…more in a bit

Example:

Message: **GOODBYE**
Key: K = 5

Encrypted message: **LTTIGDJ**
(G replaced by L, O by T, …)



To decrypt the encrypted message, move letter left 5 places

So what's the "math" behind this..

- Algorithms need to be shown to be "correct"….
- This is where the math comes in !

Some Math…the CS "discrete" math:
Circular Addition uses *Modulo arithmetic*:

(X+K) mod N = remainder of (A+K) divided by N

Ex: (6+5) mod 26 =11 *(letter L)*,
(24+5) mod 26 = 29 mod 26 = 3 *(which is letter D)*

To decrypt: (X –K) mod N
If (X-K) is negative it adds N to get result.
(3 -5) mod 26 = -2+26 = 24 = letter Y

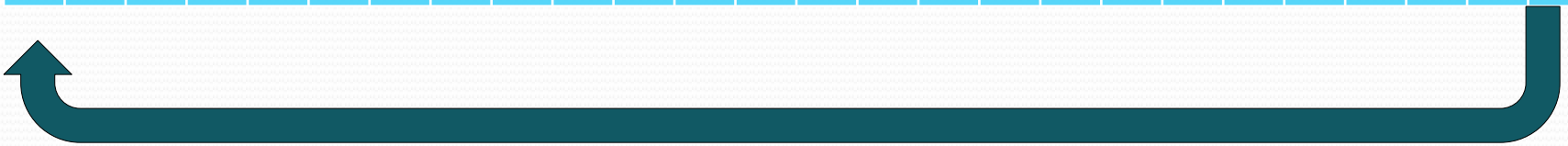
Modulo arithmetic in Python

- Circular addition
  - Circular addition…..known as Modulo
    - A Mod N = remainder of A divided by N
- Good news: Python provides the Modulo operation
  - **B = a % N**
- To encrypt value a with key K: B = (a+K)% N
  - For alphabet N=26 (we have 26 different values)

Question:

Can you ”decode” the day of the week: OQPFCA ?

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25



Weak encryption vs Strong encryption

- Strength of encryption = How easy is it to decipher your secret (i.e., encryption)
- In Caeser's cipher we use the same key for each character in our message
  - Shift each alphabet by 5
- Another method: version of One-Time-Pad (OTP)
- Encrypt each position in message with a separate key
  - Message = BYE
  - Shift B by 3, shift Y by 7, shift E by 5 to get EFJ

An application using Encryption & implementation in Python today….

- You want to send a picture to a friend
  - Or better yet, post it on a website
- To restrict who can see it, you want to encrypt it and only those with the correct key will be able to see the picture
- Steps:
  1. Take your photo
  2. Import into your Python code and enter a secret Key
  3. Write (and run) python code to encrypt the selfie
     - Implement the encryption algorithm we discussed
  4. Decrypt with the key – a wrong key will lead to a jumbled image
- Checking your encryption: Look at the encrypted image and see how similar it looks to the original image
  - The less similar it looks the “stronger” (& better) the encryption!

Getting Started…some preliminaries

- An image (i.e., your selfie) is a matrix of pixels
- To simplify our algorithm (for purpose of demonstration!) we convert your image to a grayscale image
- input image is a N by M matrix A[i,j] of pixels and key=K
  - Each pixel A[i,j] has a greyscale value between 0 and 255
    - i.e., 256 different values – analogy with 26 letters in alphabet
- To encrypt image, for each pixel add K to A[i,j] to get B[i,j]
  - Important: Circular addition with 256 different values
  - Python operator: %
    - B[i,j] = (A[i,j] + K) % 256

A better encryption "system"

- We saw how the one-time pad (OTP) is a better technique
- Applying the concept to this system: each pixel A[i,j] has its own key K[i,j]
  - And then algorithm is B[i,j] = (A[i,j] + K[i,j]) % 256
- Here is a cool trick: instead of entering gazilion values of K[i,j], how about using a 'secret' image as your key ?!!
  - Key image K, represented as a matrix K[i,j]
- Change to algorithm:
  - import the key image as K[i,j] convert to greyscale
  - B[i,j] = (A[i,j] + K[i,j]) % 256

Lessons Learned…..and CS ?

- Experience the process of going from problem to solution to “software system”
  - Your data came from different source (i.e, camera but could be a sensor in a system – camera in a car, satellite images, …?)
- Theoretical (math) basis for the solution ensures we design a correct solution
- To make this into a product, you need to implement a nice user interface or an app!
- CS is all about problem solving and then translating to implementation on a computer system
  - Lot of work in CS@GW that focuses on ”CS with Social Impact”