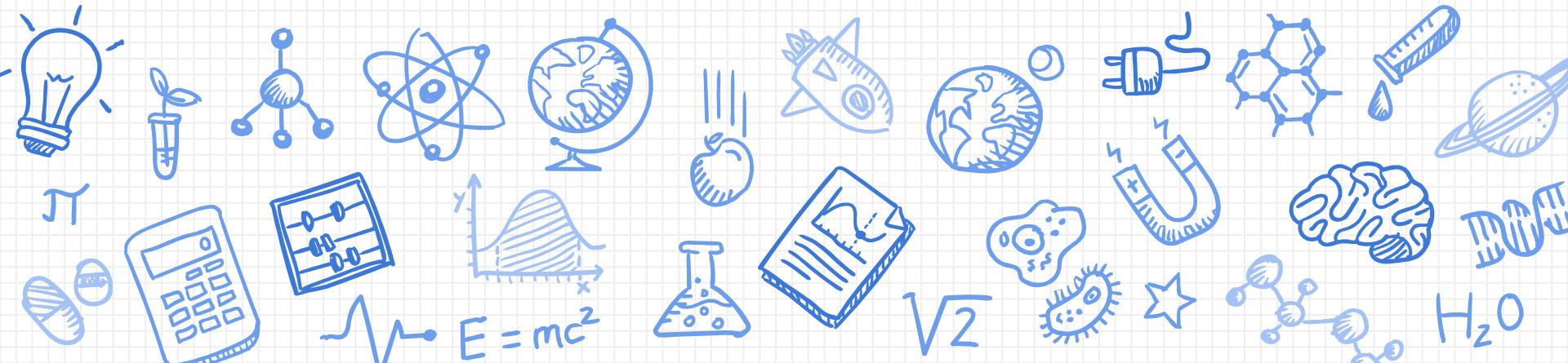
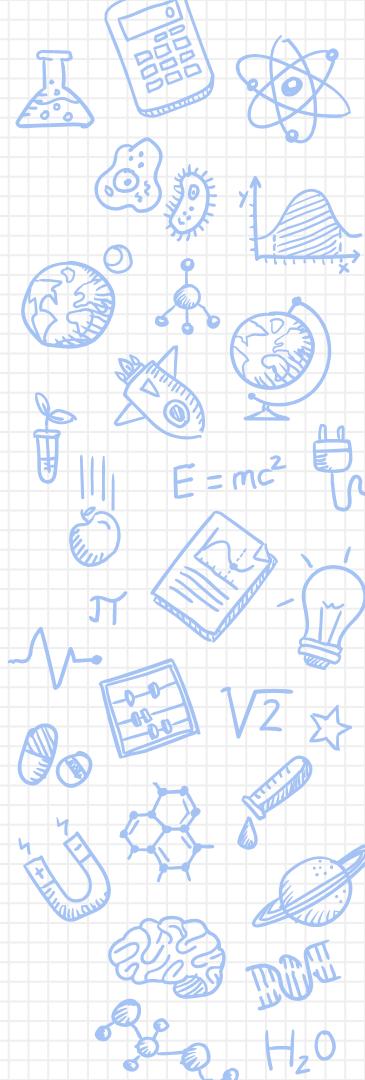


EECS16A Imaging 2



Agenda

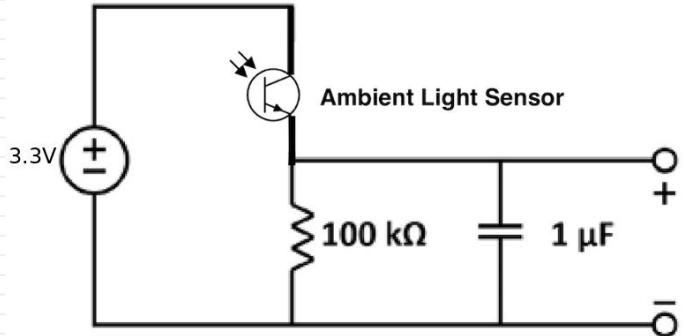
- Quick overview + review
- Images as matrices and vectors
- Pixel-by-pixel scanning
- Reconstructing scans as images
- Lab-specific simulation directions



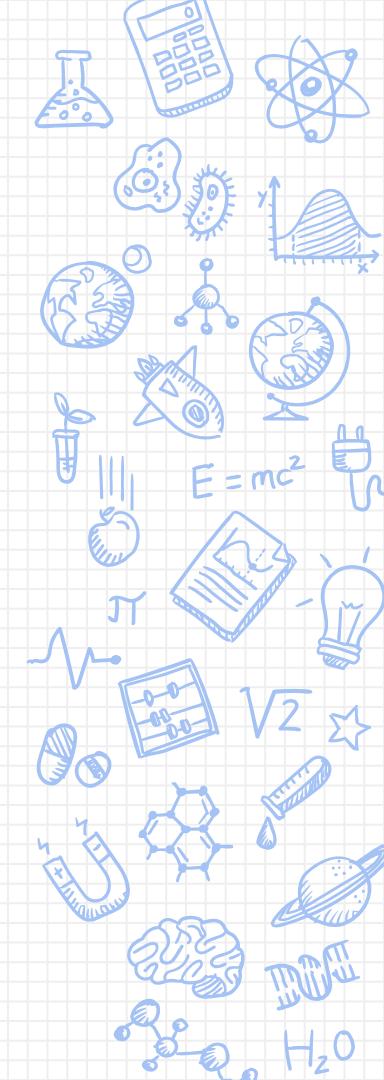
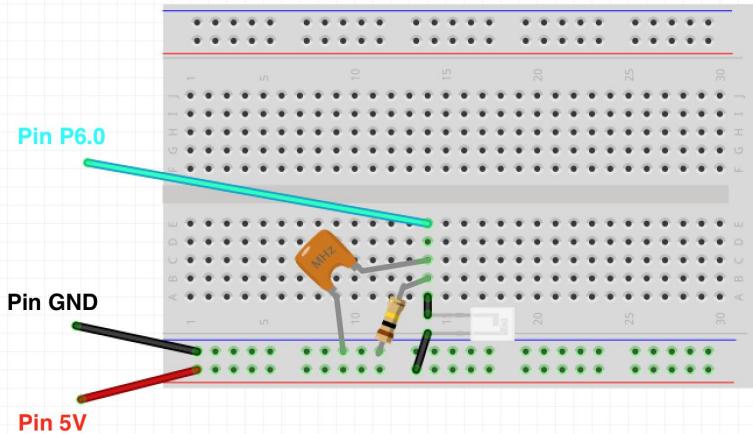
Last Week: Imaging 1

- Built our very first circuit!
 - What did this circuit do?

Circuit Diagram

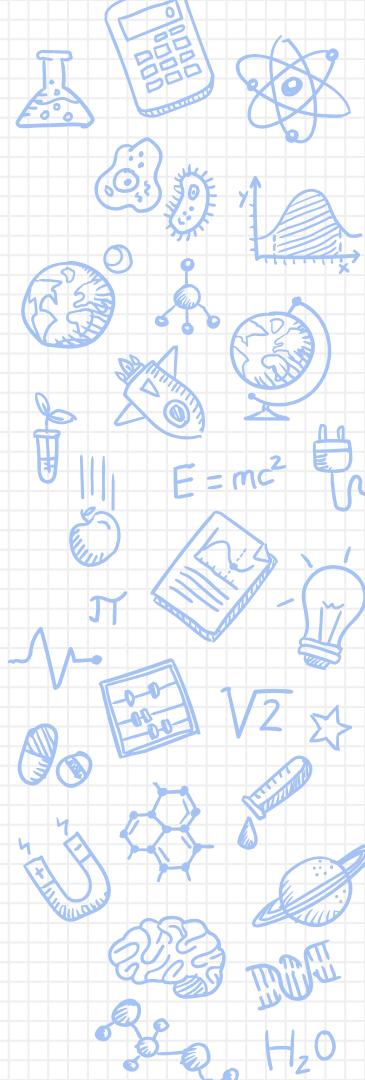


Breadboard Diagram



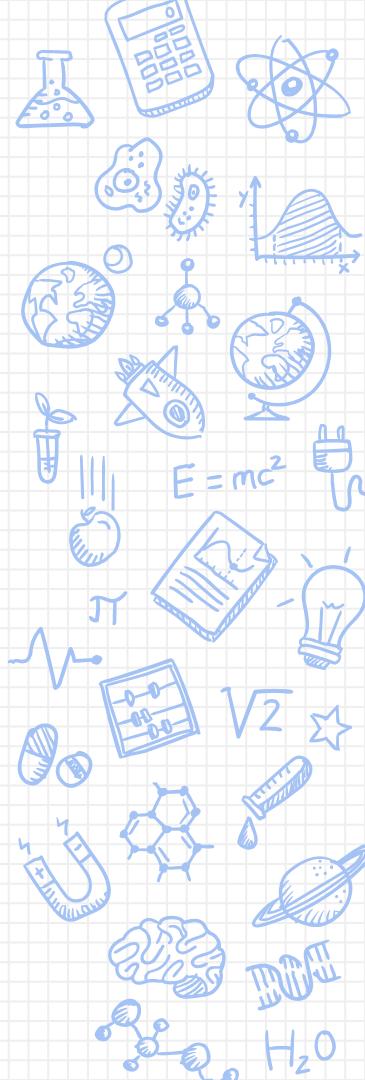
Today's Lab: Single Pixel Scanning

- Circuit from last week measures **light** intensity
 - Simulated projector illuminates image in a controlled way
 - Python programming to reconstruct image

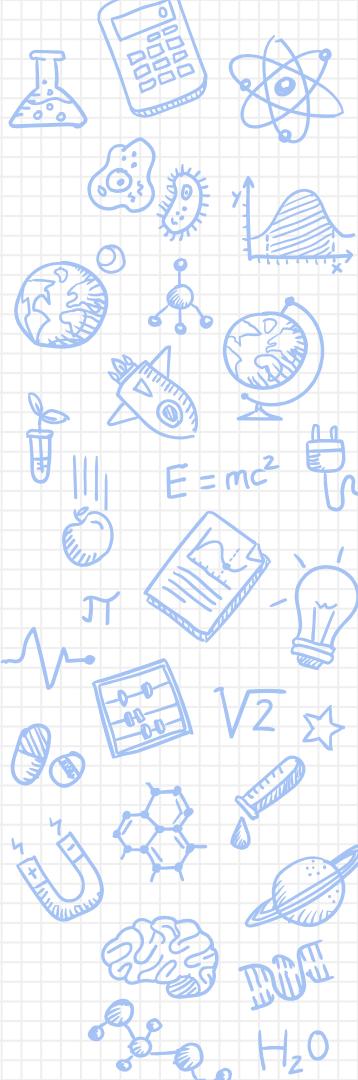


Why?

- Imaging 1:
 - Finding a link between physical quantities and voltage is powerful
 - If you can digitize it, you can do anything (IOT devices, internet, code, processing)
- Imaging 2:
 - What measurements are good measurements?
 - Remember Kody and Nara from Dis2B



Kody and Nara



2. Finding The Bright Cave

Nara the one-handed druid and Kody the one-handed ranger find themselves in dire straits. Before them is a cliff with four cave entrances arranged in a square: two upper caves and two lower caves. Each entrance emits a certain amount of light, and the two wish to find exactly the amount of light coming from each cave. Here's the catch: after contracting a particularly potent strain of ghoul fever, our intrepid heroes are only able to see the total intensity of light before them (so their eyes operate like a single-pixel camera). Kody and Nara are capable adventurers, but they don't know any linear algebra – and they need your help.

Kody proposes an imaging strategy where he uses his hand to completely block the light from two caves at a time. He is able to take measurements using the following four masks (black means the light is blocked from that cave):

Cave Labels

| | |
|-------|-------|
| x_1 | x_2 |
| x_3 | x_4 |



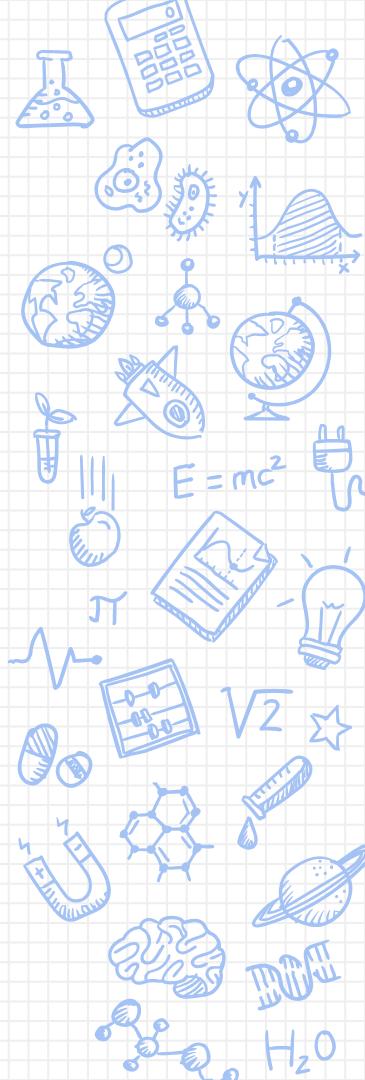
Measurement 1 Measurement 2 Measurement 3 Measurement 4

Figure 1: Four image masks.

- Let \vec{x} be the four-element vector that represents the magnitude of light emanating from the four cave entrances. Write a matrix \mathbf{K} that performs the masking process in Figure 1 on the vector \vec{x} , such that $\mathbf{K}\vec{x}$ is the result of the four measurements.
- Does Kody's set of masks give us a unique solution for all four caves' light intensities? Why or why not?
- Nara, in her infinite wisdom, places her one hand diagonally across the entrances, covering two of the cave entrances. However, her hand is not wide enough, letting in 50% of the light from the caves covered and 100% of the light from the caves not covered. The following diagram shows the percentage of light let through from each cave:

Illuminating the Big Picture

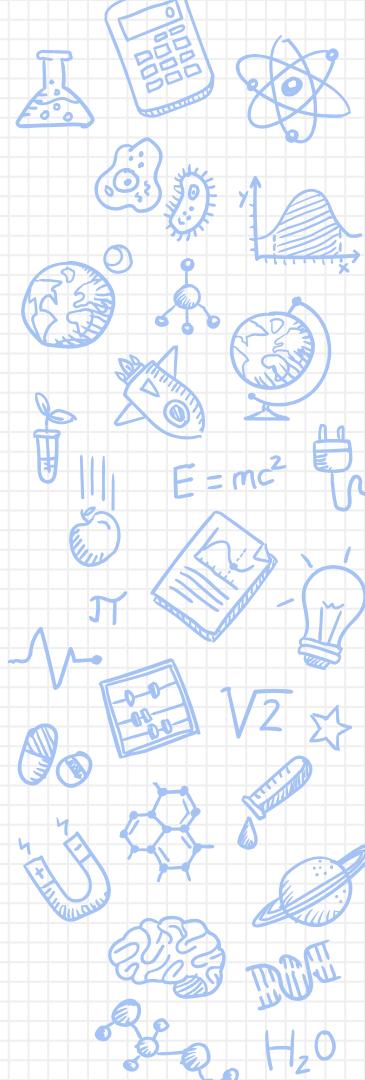
- Linear dependence
 - When can you recover your image?
 - Does it matter what mask matrix you pick?
 - Does it matter how you cover the pixels?
- Invertibility
 - When can you solve $Ax = b$?
 - How does this relate to our system?
 - How does this affect the way we pick our masking matrix?



Images, Matrices, Vectors



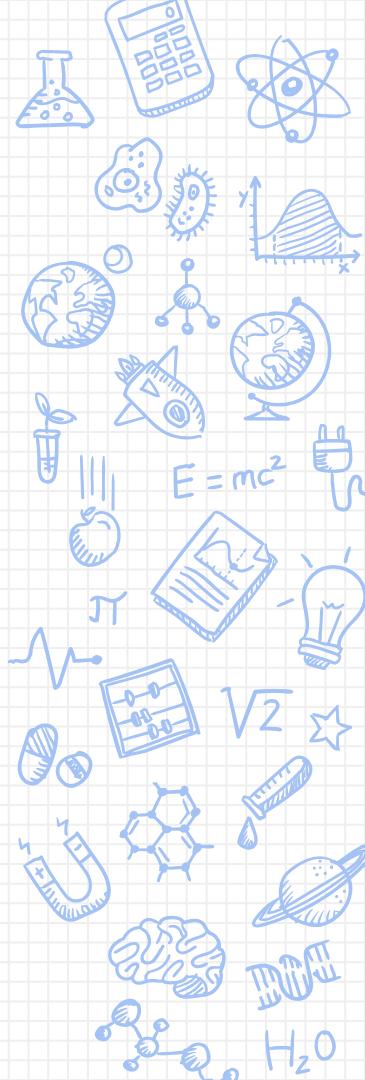
- What are the unknowns in our system?
- Want to do an experiment to get information about these unknowns
- We can do a lot of interesting processing on vectors, but we need to convert the image into one first
 - In lecture and discussion, you have seen how to turn an image into a vector. How?



Images, Matrices, Vectors



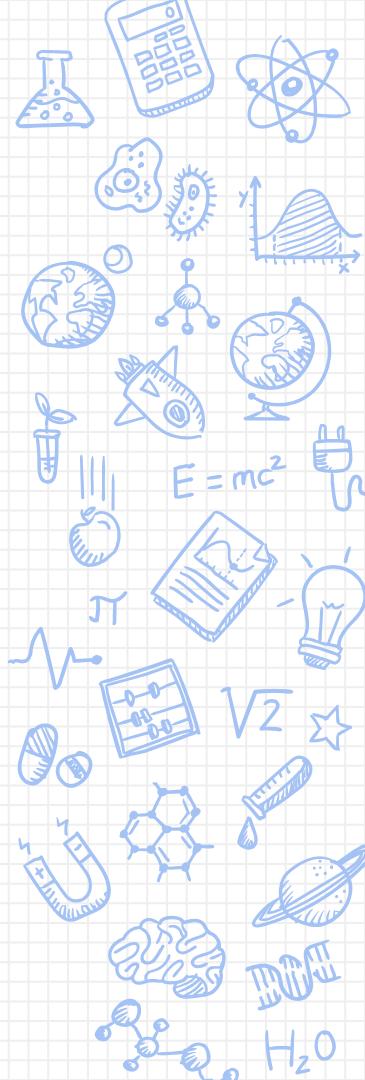
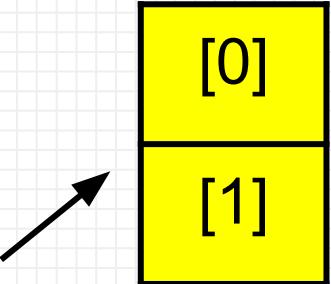
| | |
|-----|-----|
| [0] | [1] |
| [2] | [3] |
| [4] | [5] |



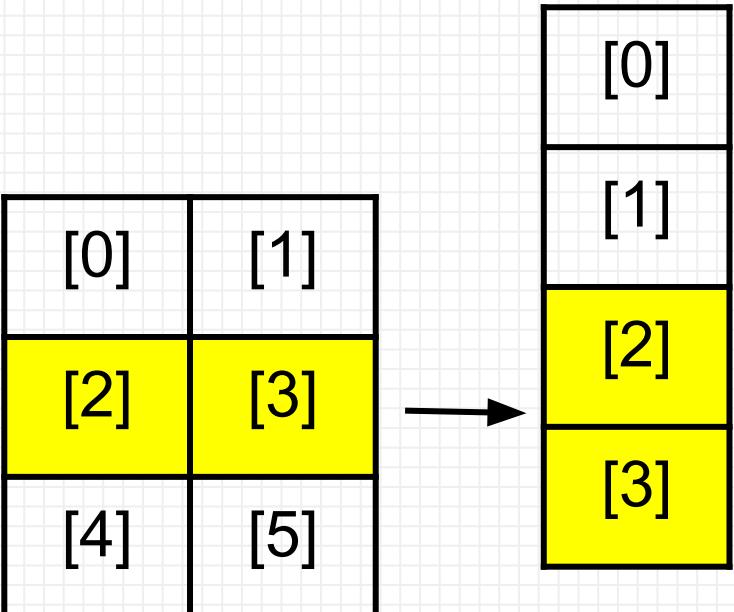
Images, Matrices, Vectors



| | |
|-----|-----|
| [0] | [1] |
| [2] | [3] |
| [4] | [5] |



Images, Matrices, Vectors

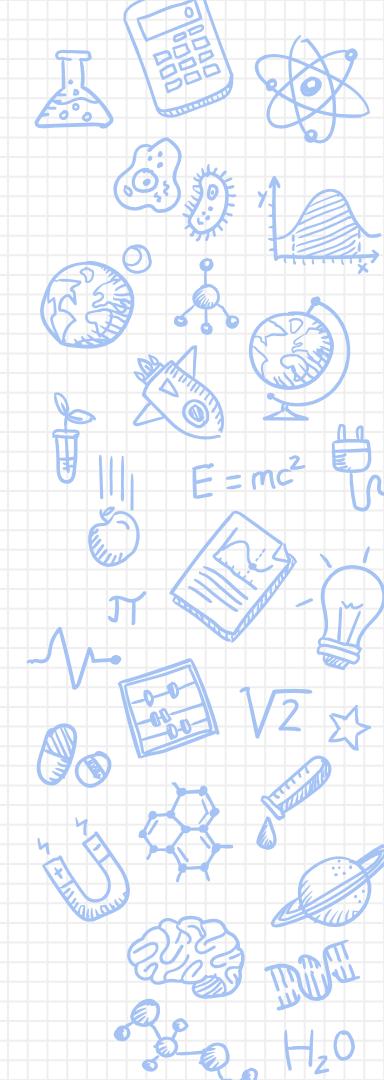


Images, Matrices, Vectors



| | |
|-----|-----|
| [0] | [1] |
| [2] | [3] |
| [4] | [5] |

| |
|-----|
| [0] |
| [1] |
| [2] |
| [3] |
| [4] |
| [5] |



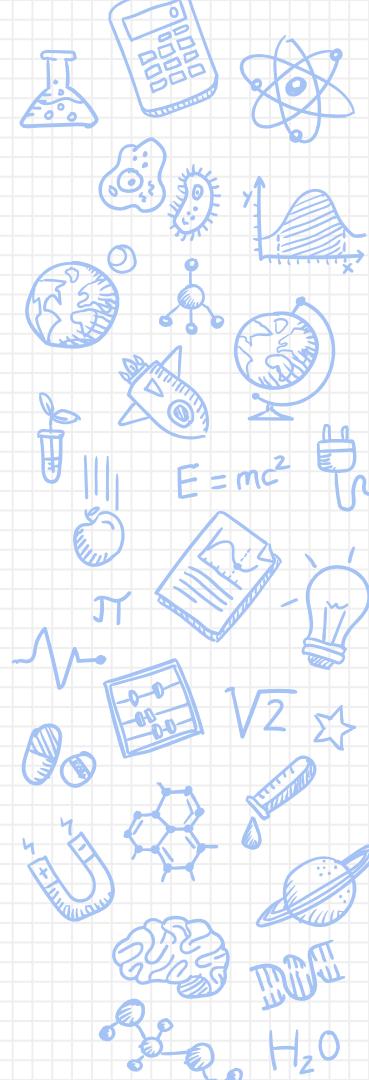
Images, Matrices, Vectors



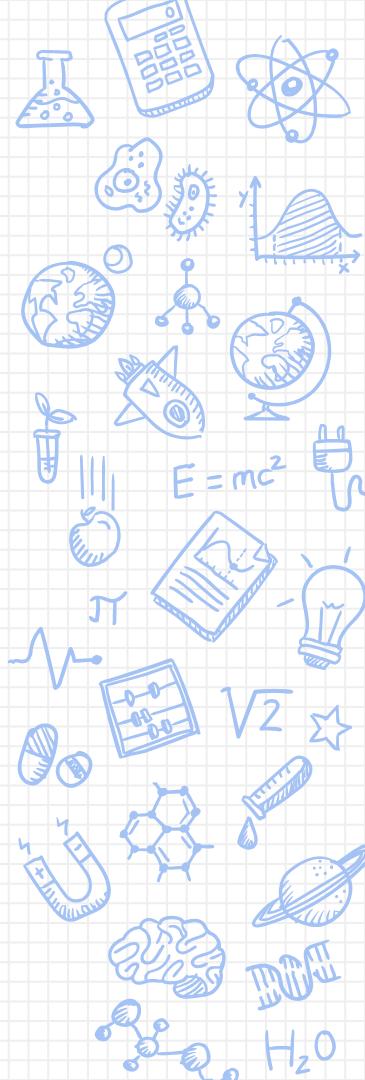
| | |
|-----|-----|
| [0] | [1] |
| [2] | [3] |
| [4] | [5] |



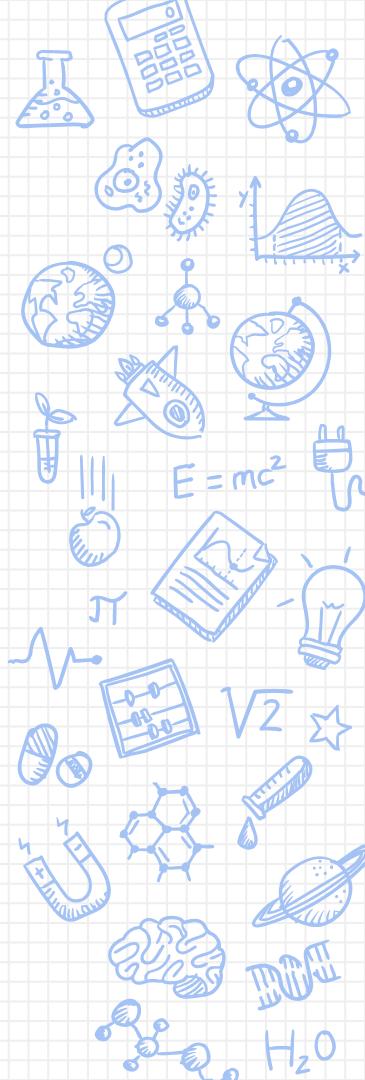
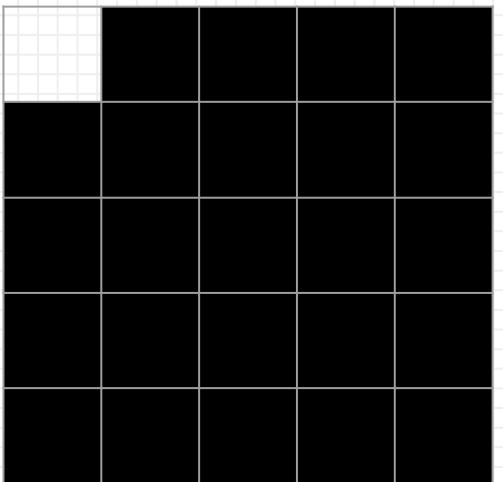
| |
|-----|
| [0] |
| [1] |
| [2] |
| [3] |
| [4] |
| [5] |



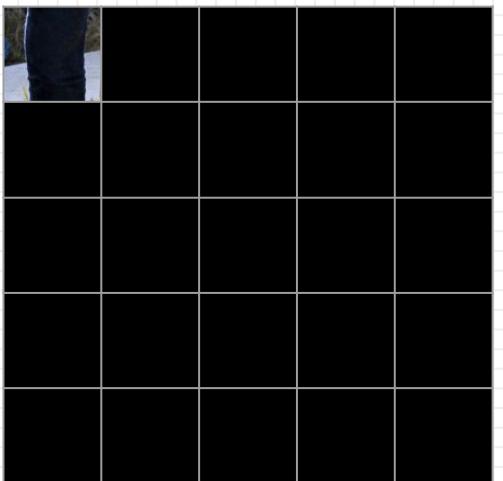
Pixel-by-Pixel Scan of an Image



Pixel-by-Pixel Scan of an Image



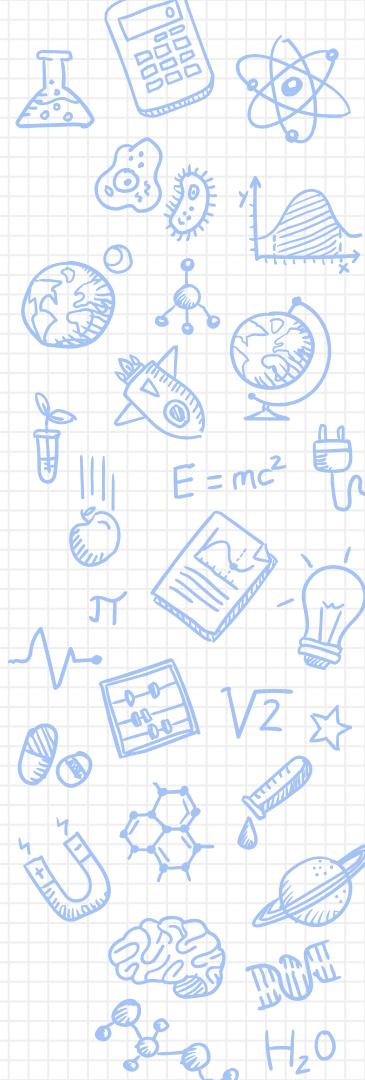
Pixel-by-Pixel Scan of an Image



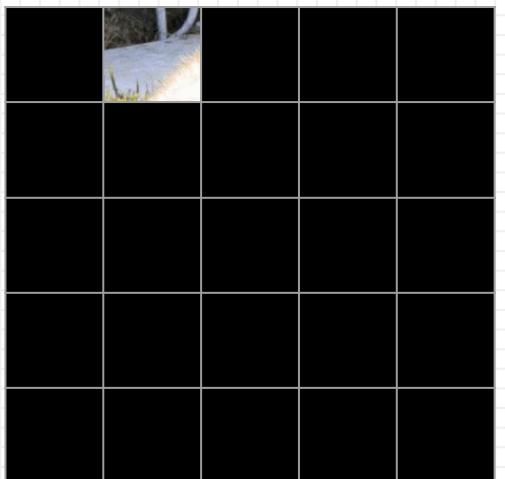
Masked image



Image



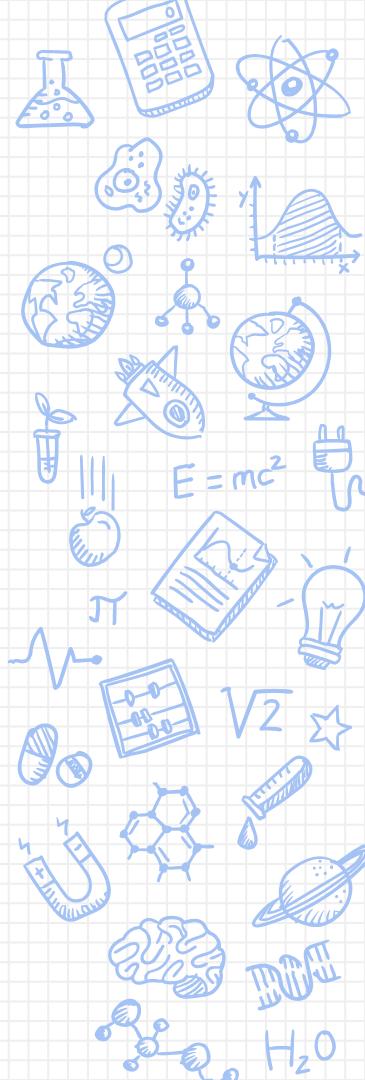
Pixel-by-Pixel Scan of an Image



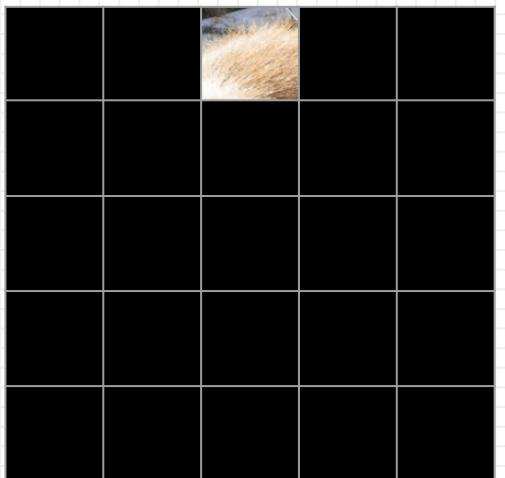
Masked image



Image



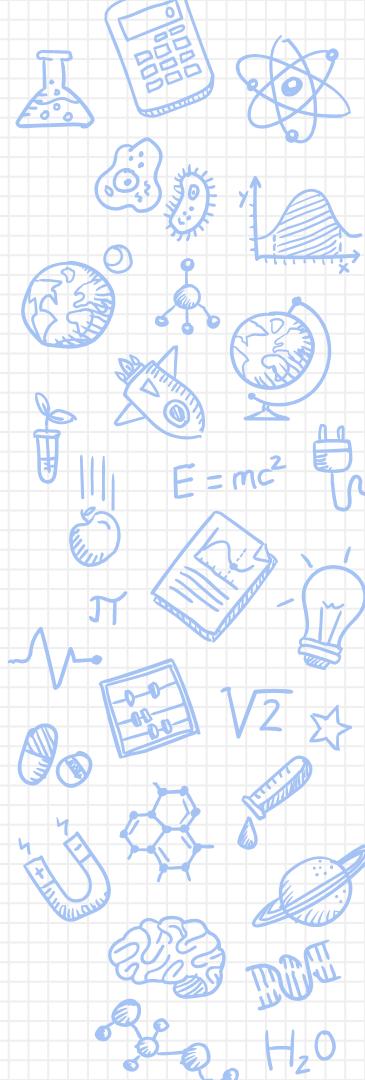
Pixel-by-Pixel Scan of an Image



Masked image



Image



Poll Time!

What would you expect the dimensions of a vector representing a 2x3 image to be?

- A. 2x3
- B. 3x2
- C. 6x1
- D. 5x1

To read all the pixels of a 4x4 image, how many pixel-by-pixel scans do we need to do?

- A. 4
- B. 8
- C. 16
- D. 32



Poll Time!

What would you expect the dimensions of a vector representing a 2x3 image to be?

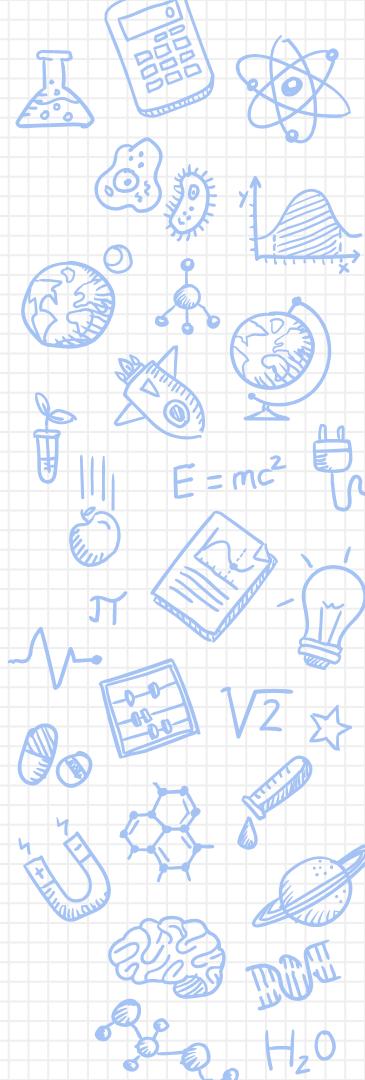
- A. 2x3
- B. 3x2
- C. 6x1
- D. 5x1

To read all the pixels of a 4x4 image, how many pixel-by-pixel scans do we need to do?

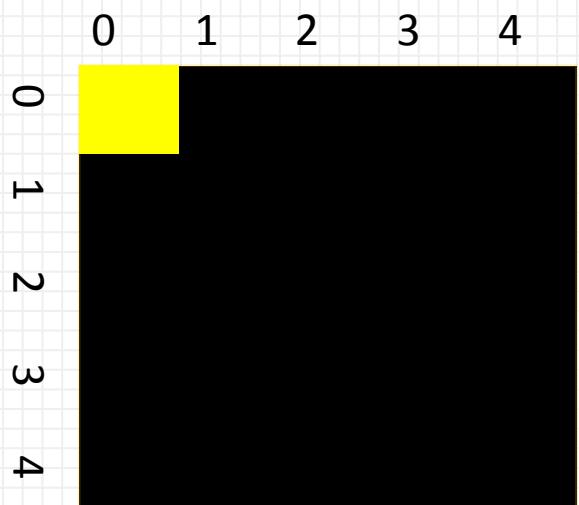
- A. 4
- B. 8
- C. 16
- D. 32



Representing our Masks in Python

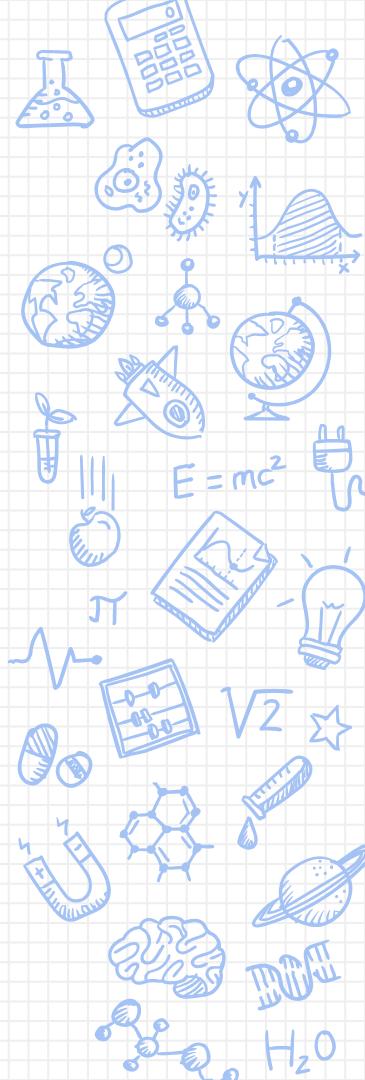


Imaging Mask 0

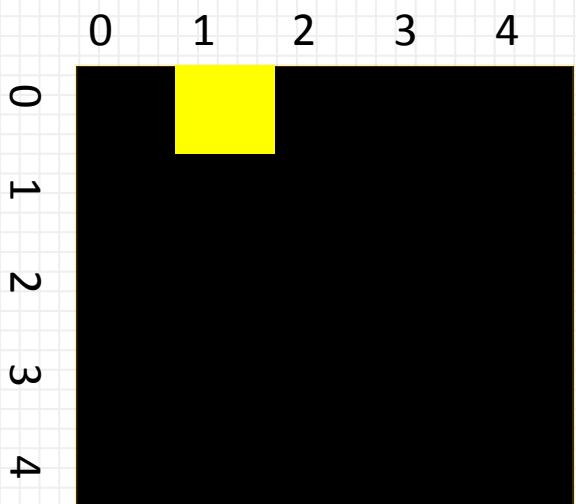


mask0 =
np.array([[1, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]])

Representing our Masks in Python



Imaging Mask 1

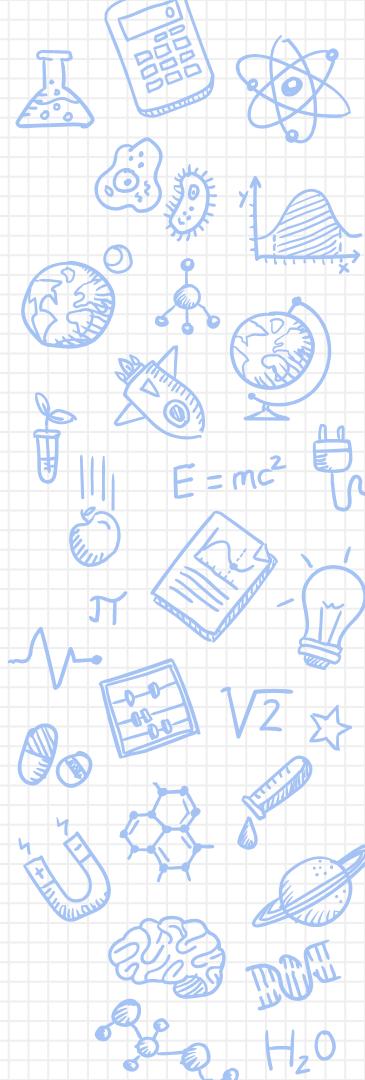
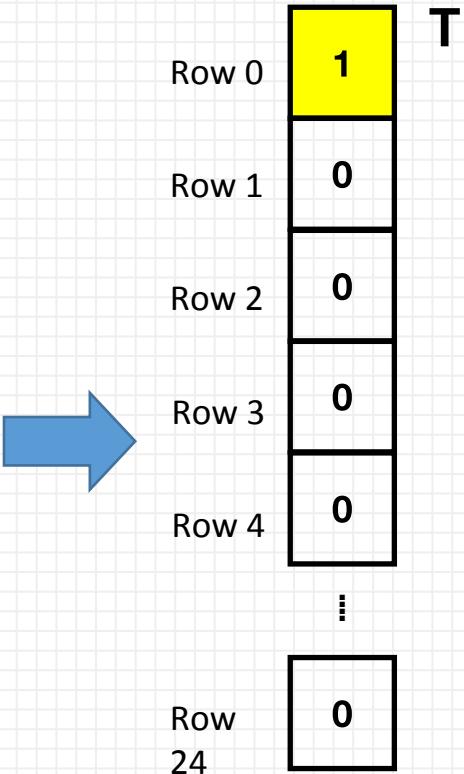


mask1 =
`np.array([[0, 1, 0, 0, 0]
 , [0, 0, 0, 0, 0]
 , [0, 0, 0, 0, 0]
 , [0, 0, 0, 0, 0]
 , [0, 0, 0, 0, 0]])`

Turning the Masks Into Vectors

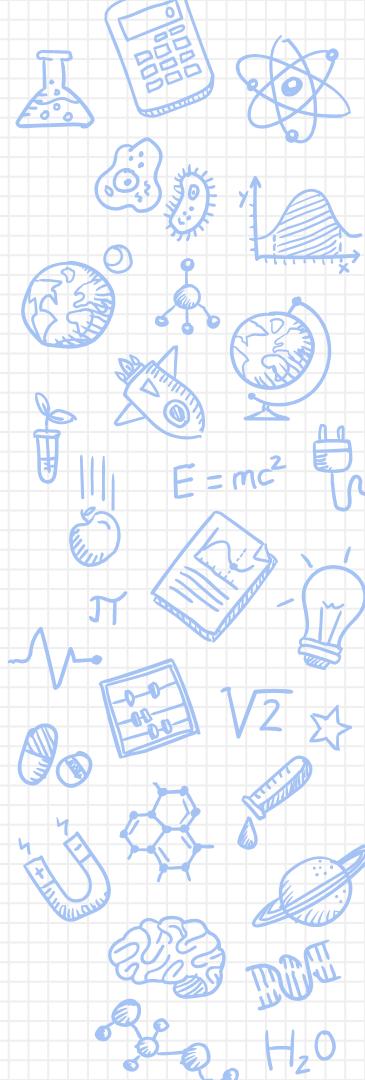
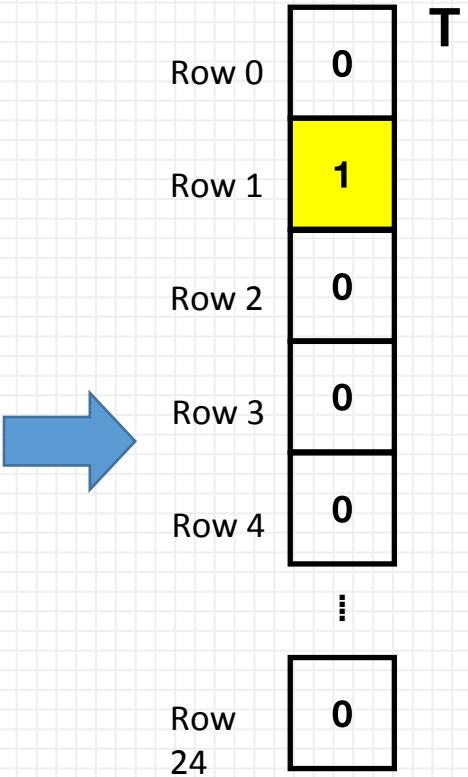
5x5 mask to 25x1 vector

```
mask0 = [[ 1, 0, 0, 0, 0  
          [ 0, 0, 0, 0, 0  
          [ 0, 0, 0, 0, 0  
          [ 0, 0, 0, 0, 0  
          [ 0, 0, 0, 0, 0 ]]]]
```



Turning the Masks Into Vectors

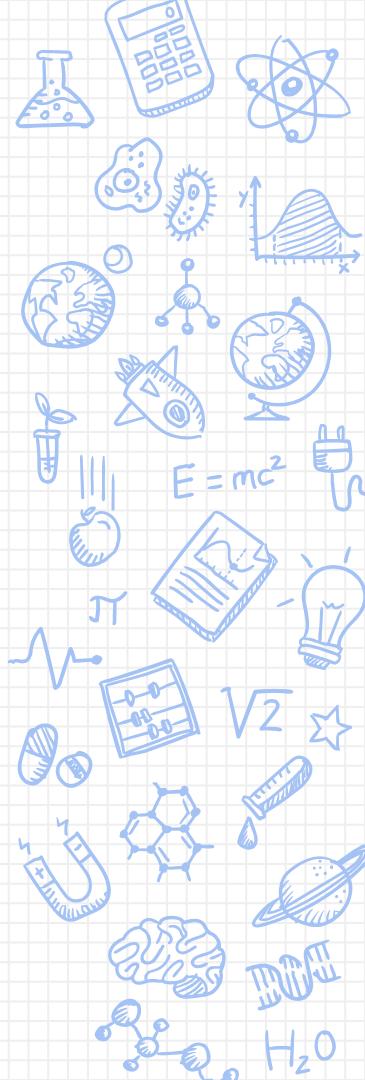
```
mask1 = [[0, 1, 0, 0, 0],  
         [0, 0, 0, 0, 0],  
         [0, 0, 0, 0, 0],  
         [0, 0, 0, 0, 0],  
         [0, 0, 0, 0, 0]]
```



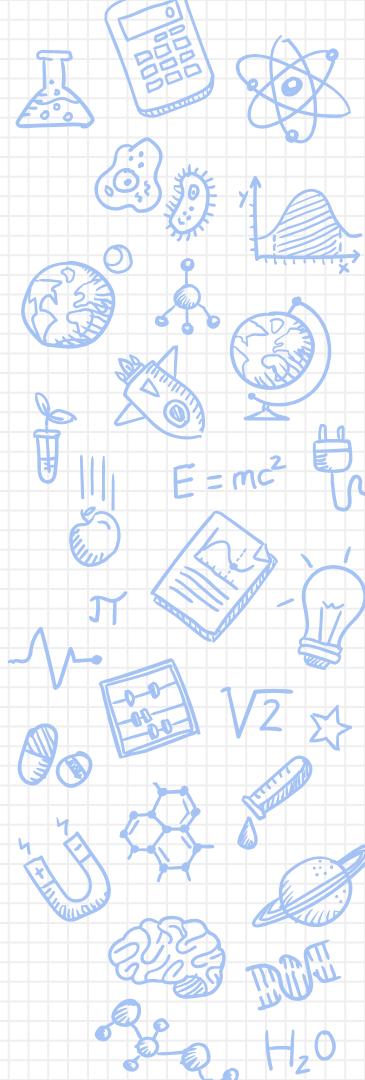
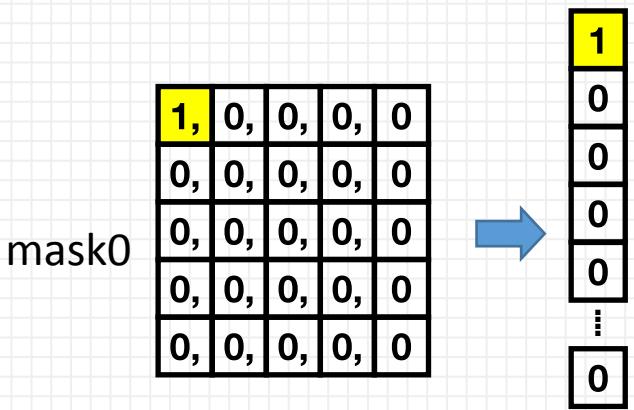
Generating the Masking Matrix from the Masks

mask0

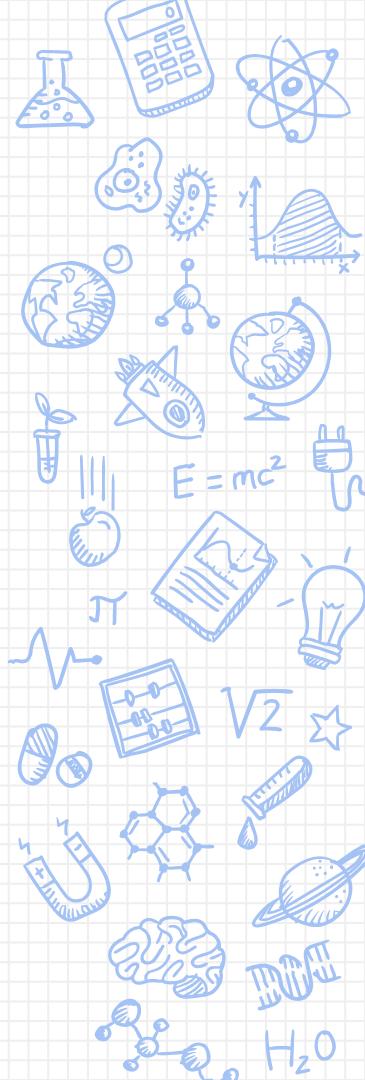
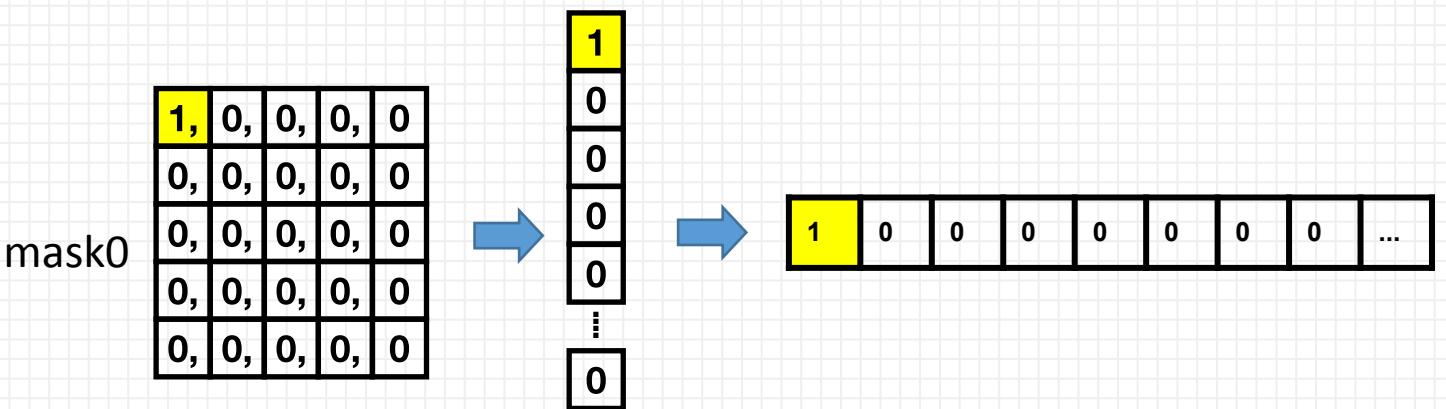
| |
|---------------|
| 1, 0, 0, 0, 0 |
| 0, 0, 0, 0, 0 |
| 0, 0, 0, 0, 0 |
| 0, 0, 0, 0, 0 |
| 0, 0, 0, 0, 0 |



Generating the Masking Matrix from the Masks



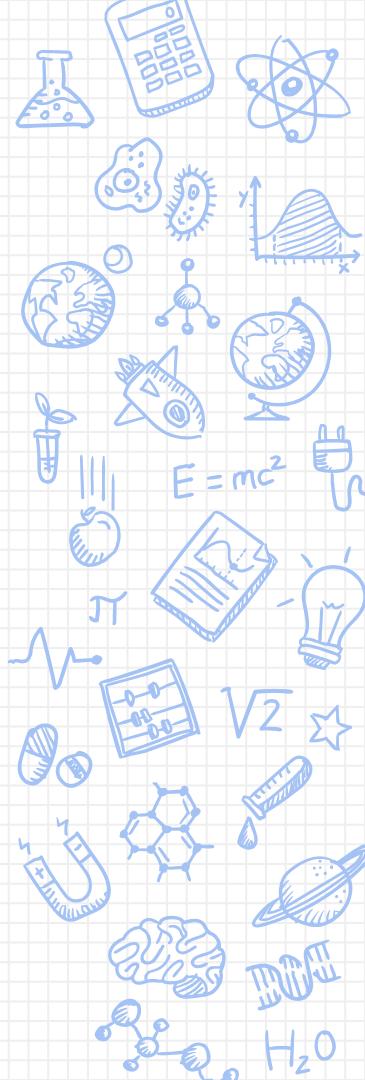
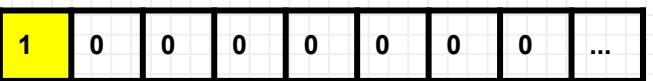
Generating the Masking Matrix from the Masks



Generating the Masking Matrix from the Masks

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

mask1



Generating the Masking Matrix from the Masks

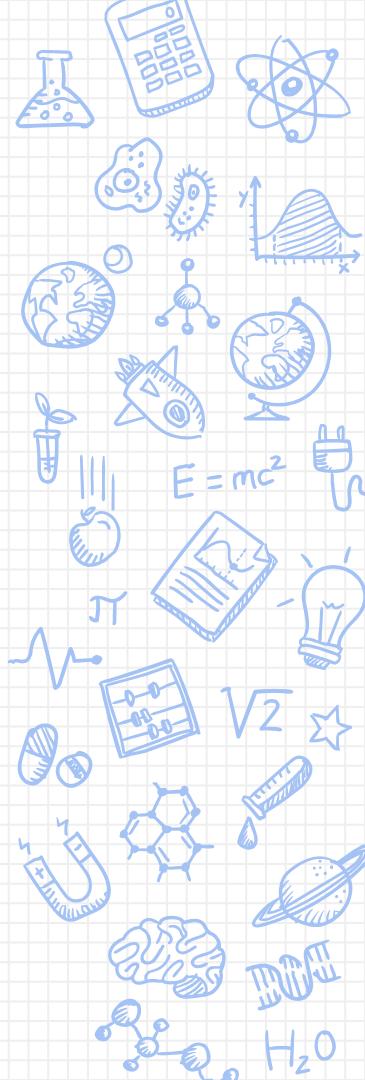
mask1

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

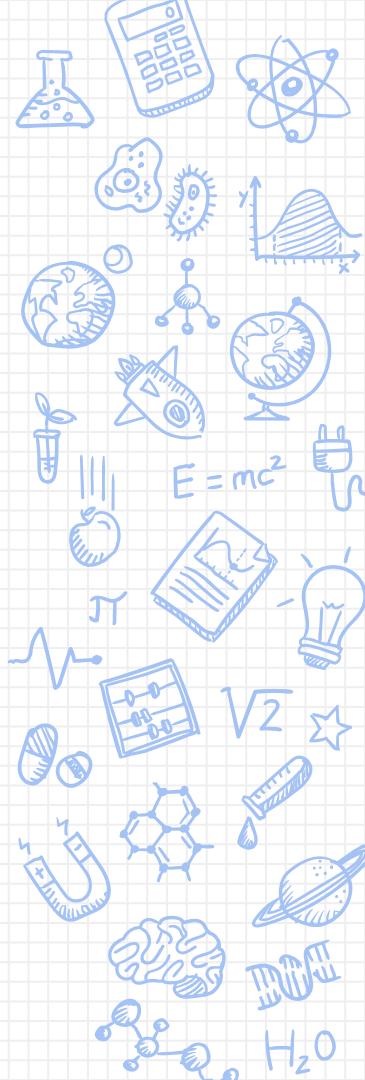
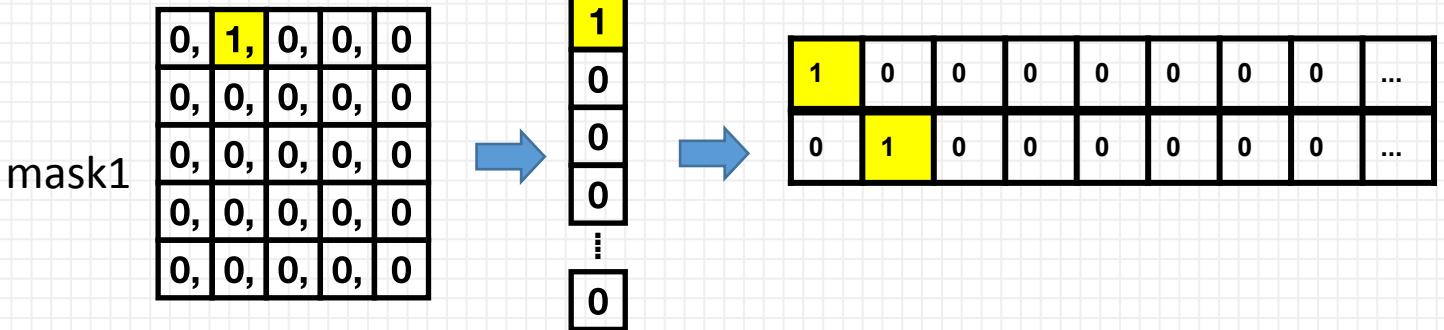


| |
|---|
| 0 |
| 1 |
| 0 |
| 0 |
| 0 |
| ⋮ |
| 0 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
|---|---|---|---|---|---|---|---|-----|

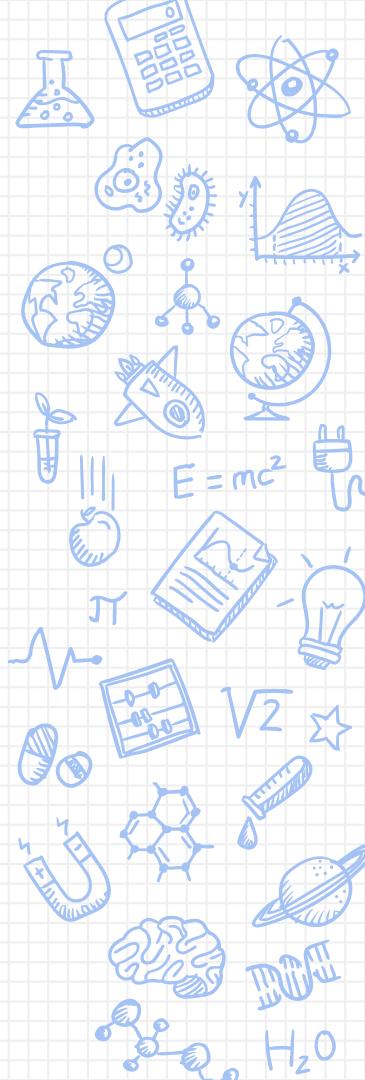


Generating the Masking Matrix from the Masks



Generating the Masking Matrix from the Masks

| | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... |

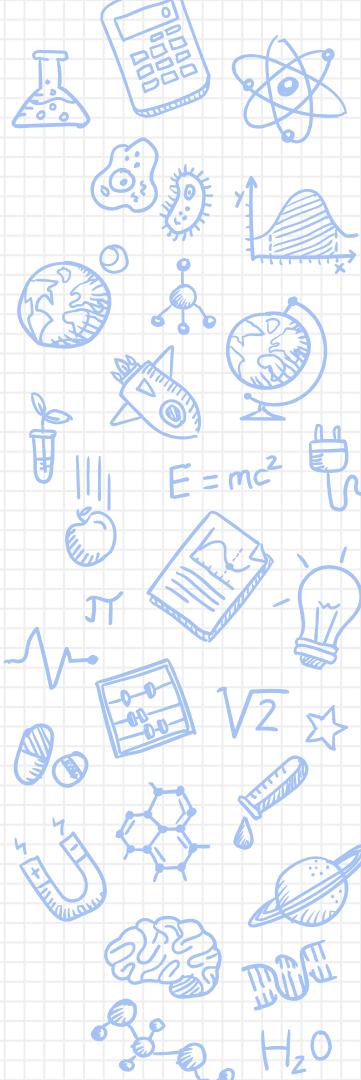


Generating the Masking Matrix from the Masks

| | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | ... |

Each Row is a Different Experiment

H =



Measuring a Pixel is Matrix-Vector Multiplication

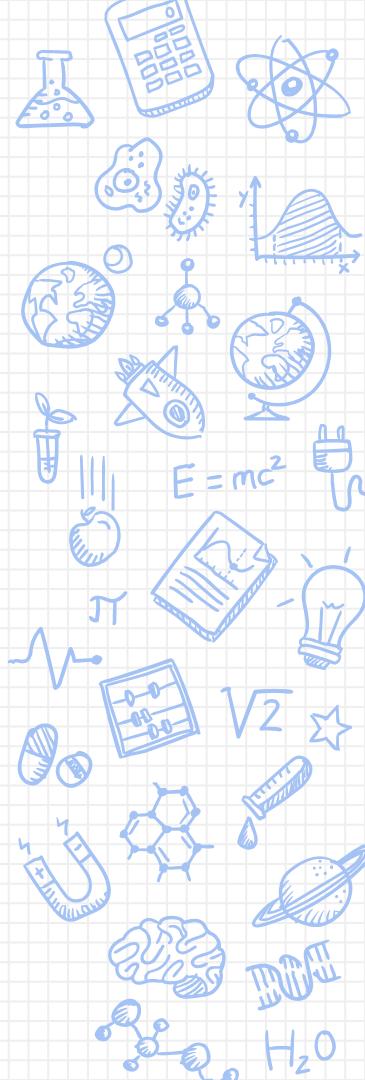
| | | | | | | | | |
|-----|---|---|---|---|---|---|---|-----|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | ... |
| ... | | | | | | | | |

Masking Matrix \mathbf{H}

$$\begin{matrix} i_1 \\ i_2 \\ i_3 \\ \vdots \\ i_n \end{matrix} = \begin{matrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_n \end{matrix}$$

Unknown,
vectorized
image, \vec{i}

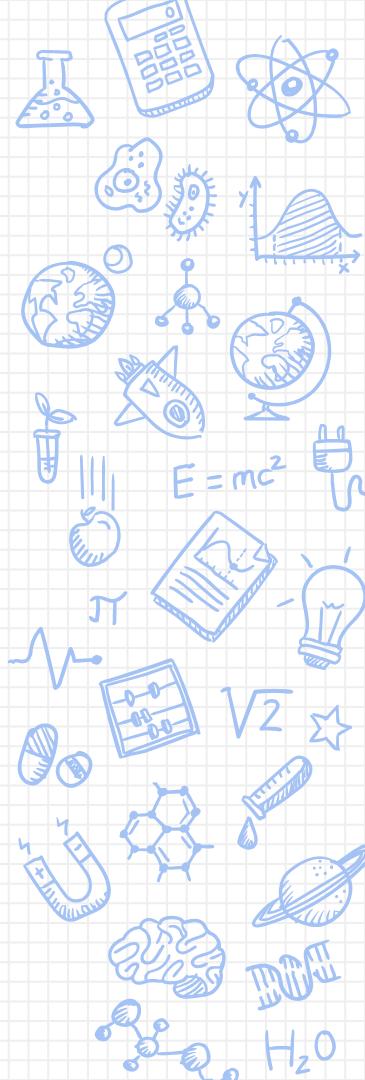
Recorded
Sensor
readings, \vec{s}



Measuring a Pixel is Matrix-Vector Multiplication

$$\vec{s} = H\vec{i}$$

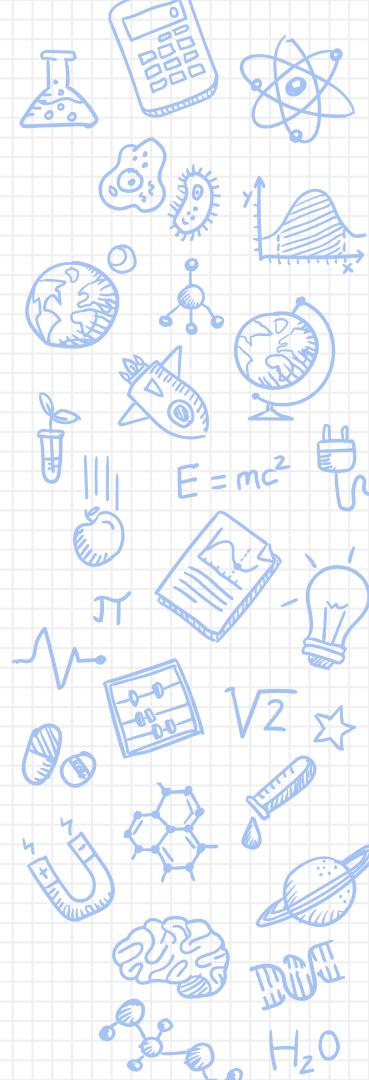
- We know H and we have the sensor readings, how do we get the image?
- How do we solve this?
- When can we solve this?
 - Conditions on H



Poll Time!

Select all of the following that must be true for the image vector i to be recoverable from the sensor vector s .

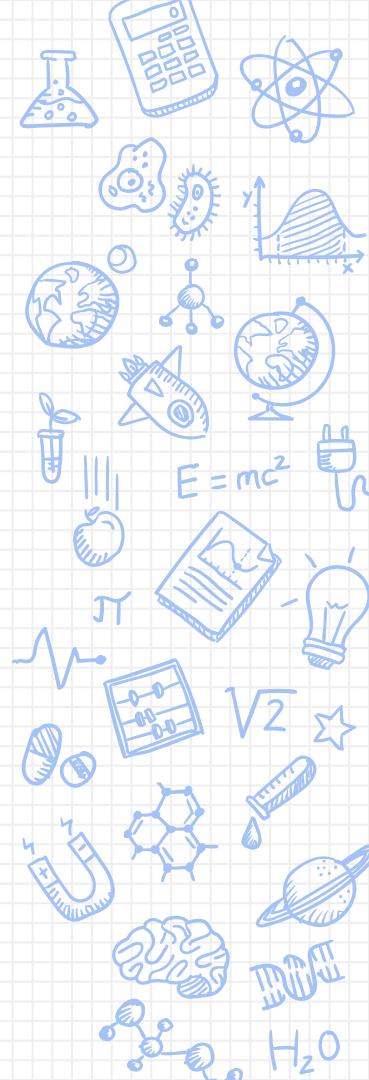
1. H must be invertible
2. H must have linearly independent rows
3. H must be a square matrix
4. H must be the identity matrix



Poll Time!

Select all of the following that must be true for the image vector i to be recoverable from the sensor vector s .

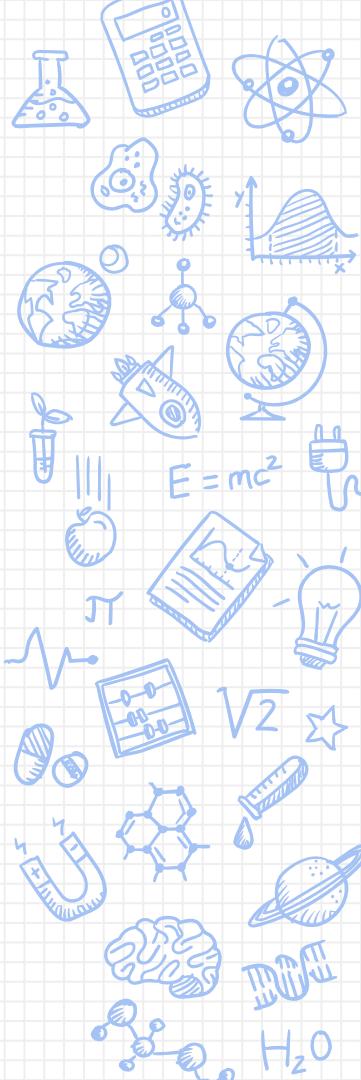
1. H must be invertible
2. H must have linearly independent rows
3. H must be a square matrix¹
4. H must be the identity matrix



How Scanning Works: iPython

H =

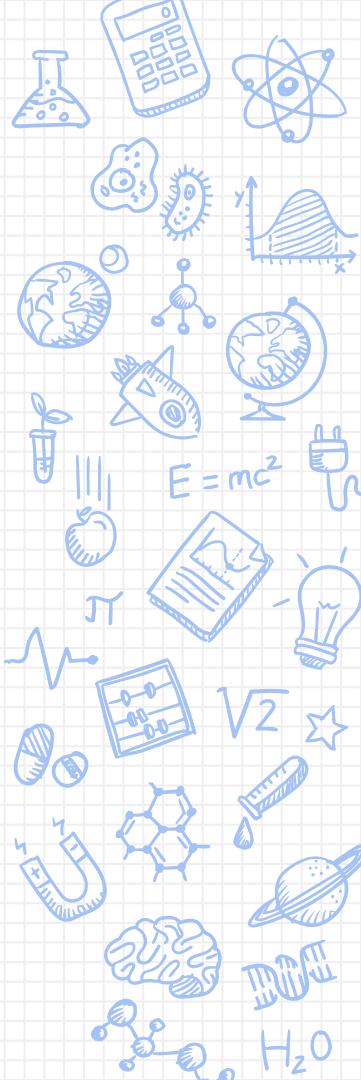
| | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | ... |



How Scanning Works: iPython

H =

| | | | | | | | | |
|---|---|---|---|---|---|---|---|-----|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | ... |



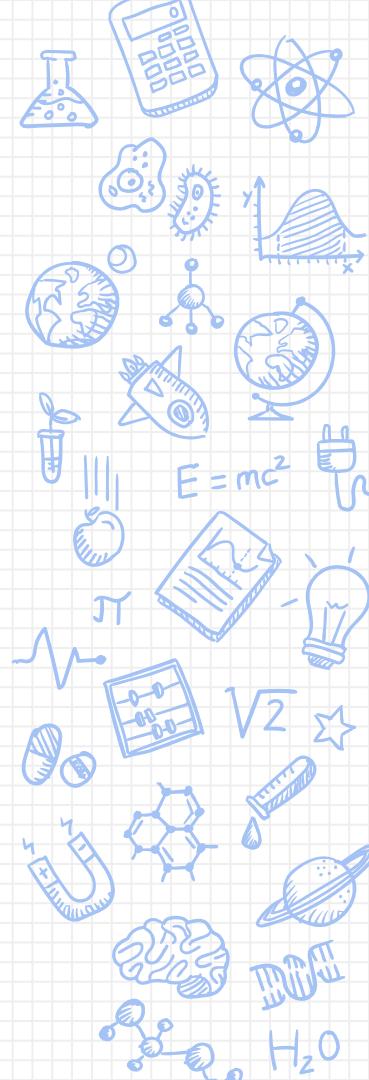
How Scanning Works: iPython

H =

| | | | | | | | | |
|-----|---|---|---|---|---|---|---|-----|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | ... |
| ... | | | | | | | | |



| |
|-----|
| 0 |
| 1 |
| 0 |
| 0 |
| 0 |
| 0 |
| 0 |
| ... |



How Scanning Works: iPython

H =

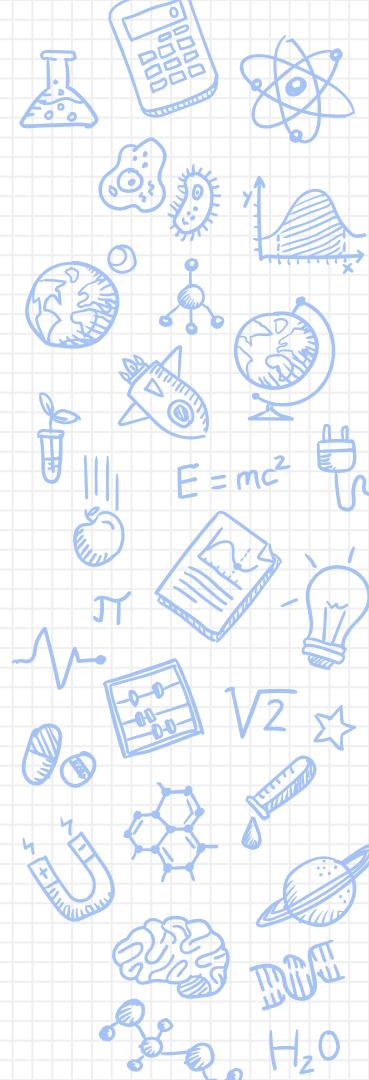
| | | | | | | | | |
|-----|---|---|---|---|---|---|---|-----|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | ... |
| ... | | | | | | | | |



| |
|-----|
| 0 |
| 1 |
| 0 |
| 0 |
| 0 |
| 0 |
| ... |
| 0 |



| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |



How Scanning Works: iPython

H =

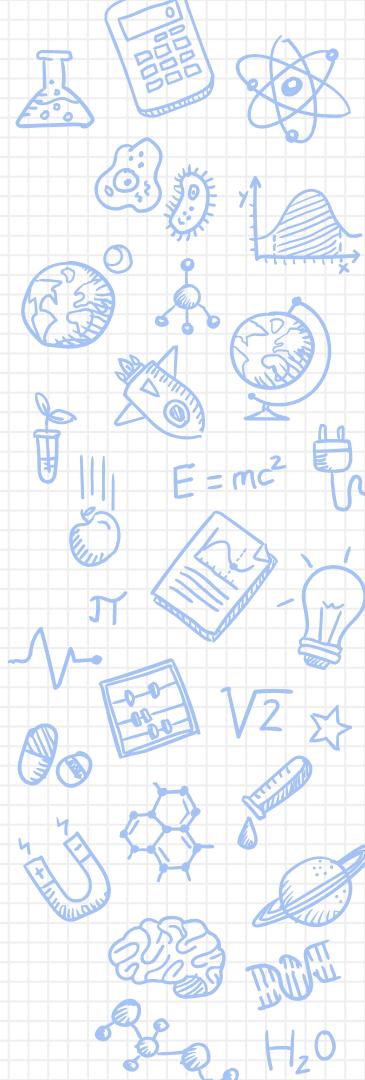
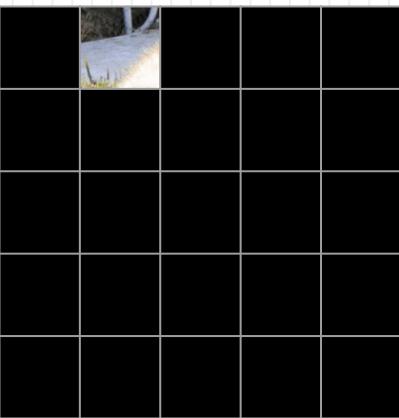
| | | | | | | | | |
|-----|---|---|---|---|---|---|---|-----|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | ... |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | ... |
| ... | | | | | | | | |



| |
|-----|
| 0 |
| 1 |
| 0 |
| 0 |
| 0 |
| 0 |
| ... |
| 0 |

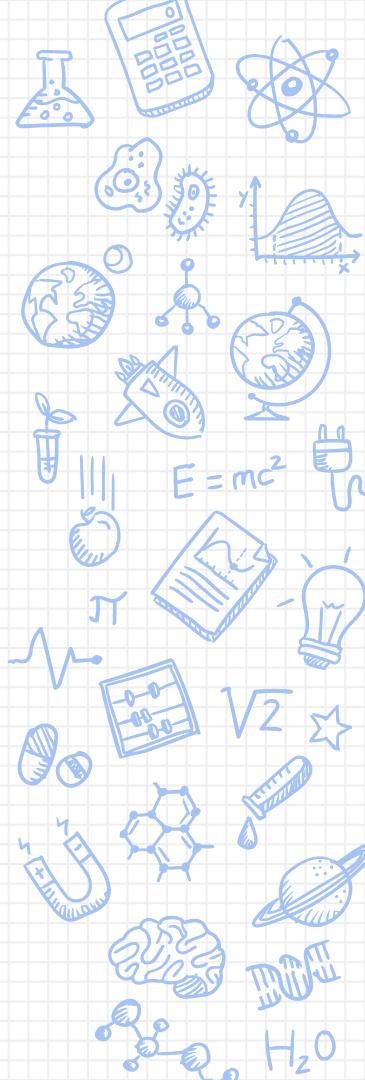


| | | | | |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |



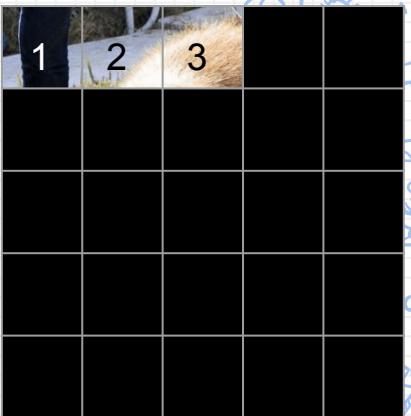
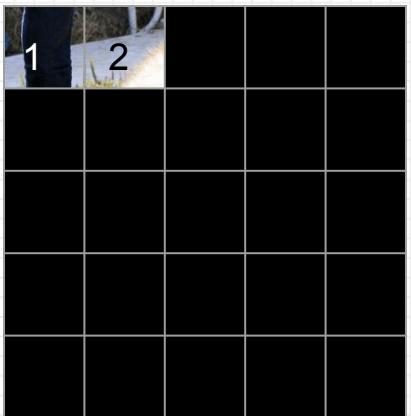
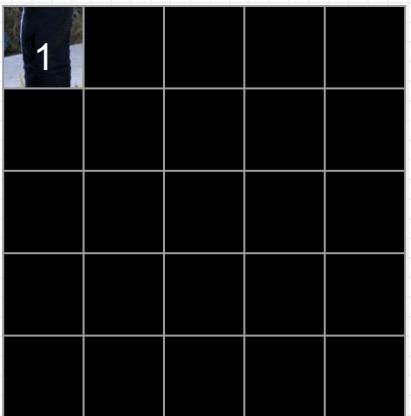
What Makes a Mask Good?

- Linearly independent columns → Invertible
 - Can't get a solution without this
 - There is a unique solution
- What would be a bad mask?
- Food for thought: Are all invertible matrices equally as good?
 - Find out in Imaging 3 next week

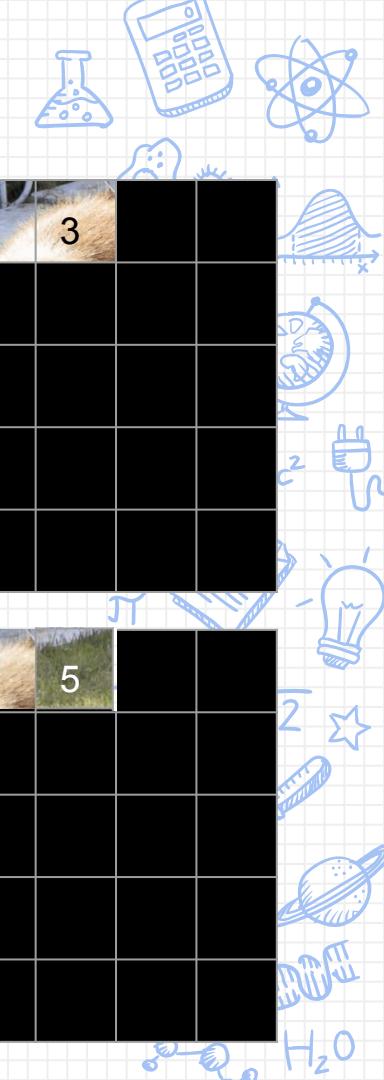
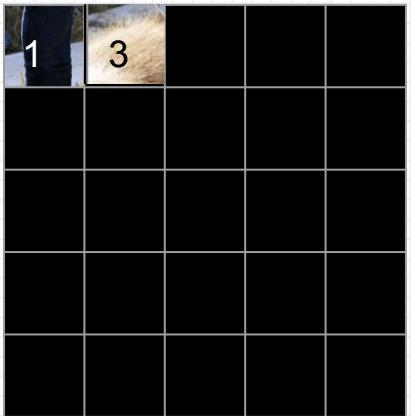
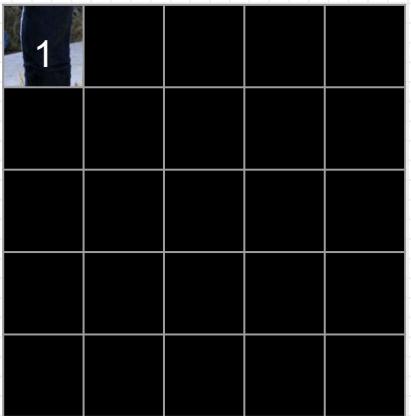


Cumulative Scanning

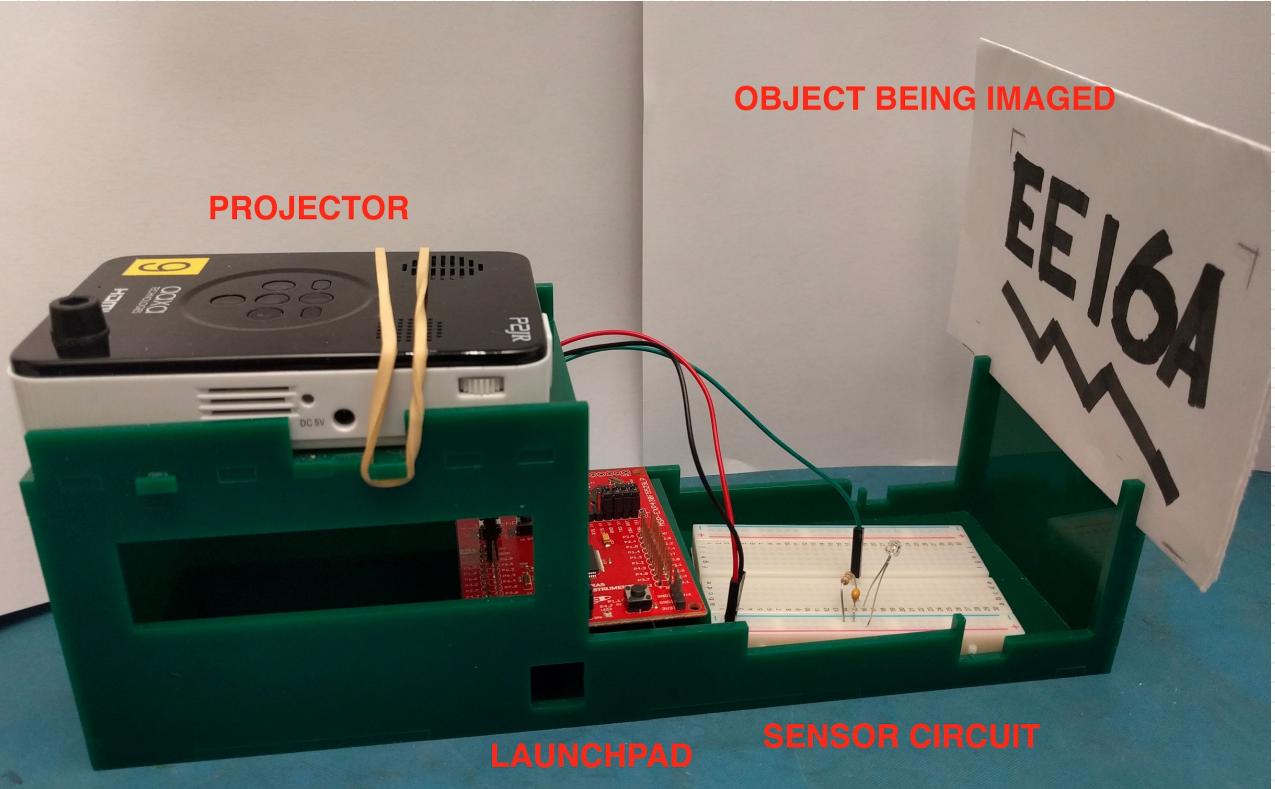
Identity
(H)



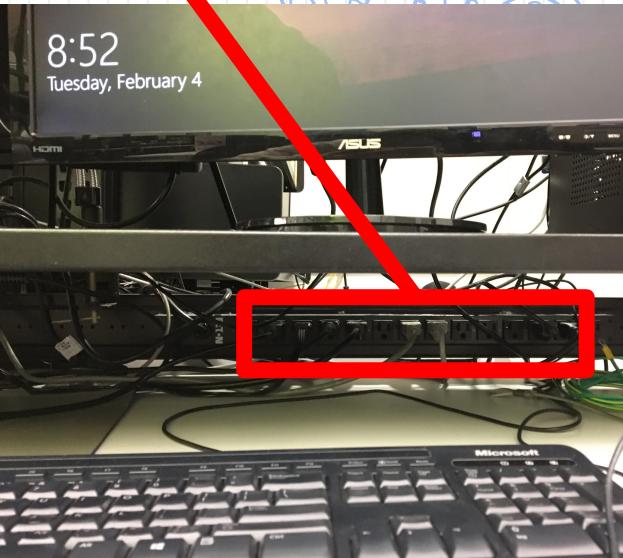
Alternating
(H_Alt)



Real-life Setup

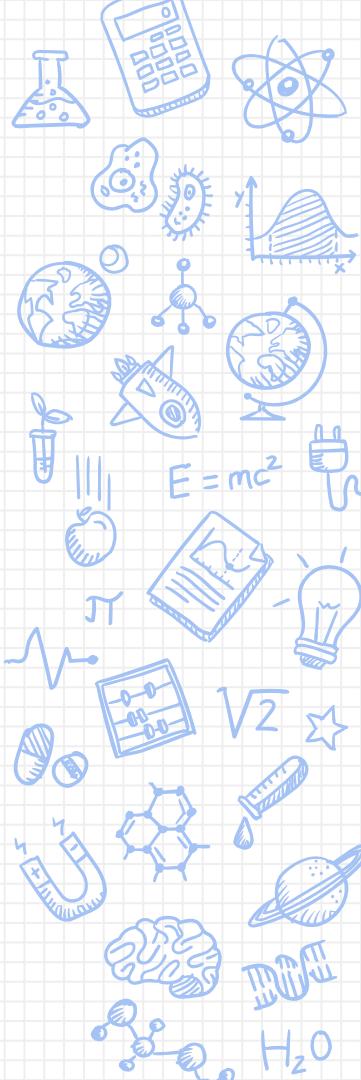


Power strip to power your projector



This Lab: Software Simulator Setup

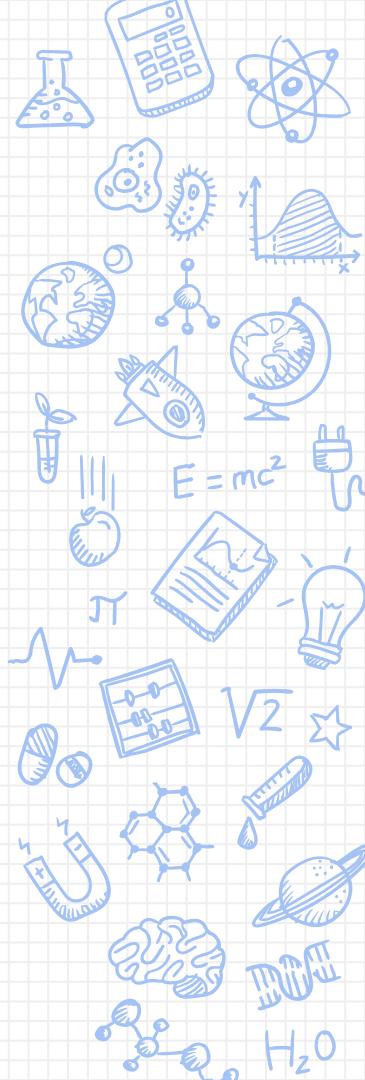
1. Upload (“simple”) image of an object (or use default)
 2. Shrink your image (preferably 32x32)
 3. ‘Project’ masks (rows of H) onto it and “measure” s using matrix multiplication
 4. Multiply by H inverse to find i ($= H^{-1}s$)



Poll Time!

Select all of the following that describe the relationship between H (the masking matrix), s (the sensor vector), and i (the image vector)?

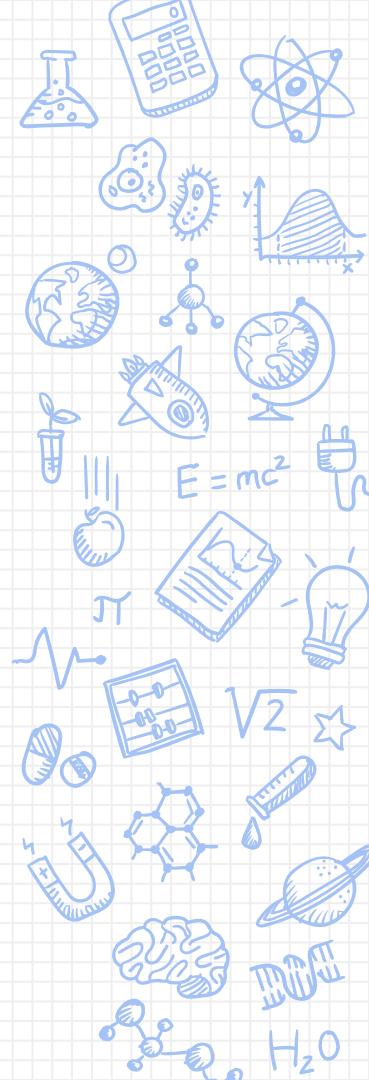
1. $Hs = i$
2. $Hi = s$
3. $H^{-1}i = s$
4. $H^{-1}s = i$
5. $i * s = H$



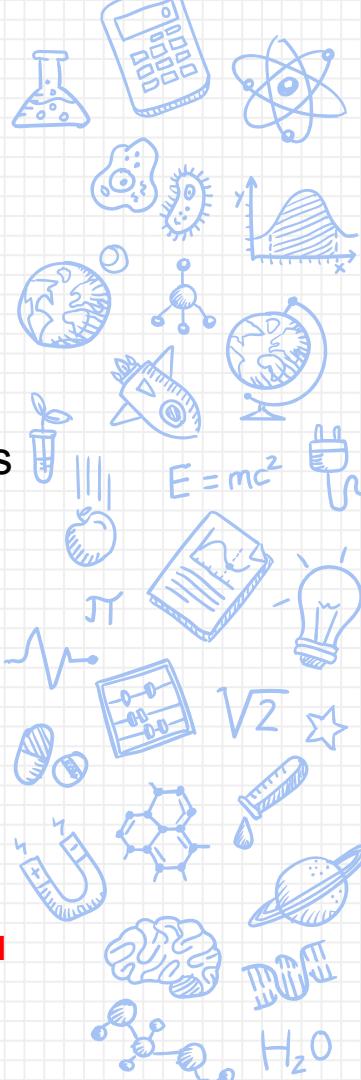
Poll Time!

Select all of the following that describe the relationship between H (the masking matrix), s (the sensor vector), and i (the image vector)?

1. $Hs = i$
2. $Hi = s$
3. $H^{-1}i = s$
4. $H^{-1}s = i$
5. $i * s = H$



Important Notes



1. Pick a simple image! Quality can be lost when resizing.
2. Use a short / simple imagePath name
 - a. you'll have to fill this in for some cells
 - b. default is the home directory for the lab
3. Before starting the simulations, open up the link (in the directions above the code cell) to the display view in a different tab & observe as the cell is running
4. Read simulation descriptions carefully!
 - a. you might have to manually set height & width because the defaults are 32x32
5. Each mask section includes ideal + noisy imaging
 - a. noisy imaging meant to simulate real projector behavior
 - b. don't worry about this for now -- more to come in Img 3
6. **Post in the Week 3 Lab Announcements Piazza thread if you still have an unresolved Energia error from Imaging 1**