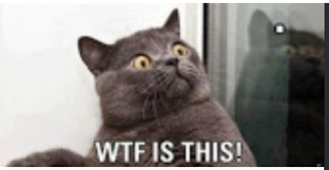
ENR145 Computational Methods:
A small detour from “Hamming Python” so we can do more coding practice

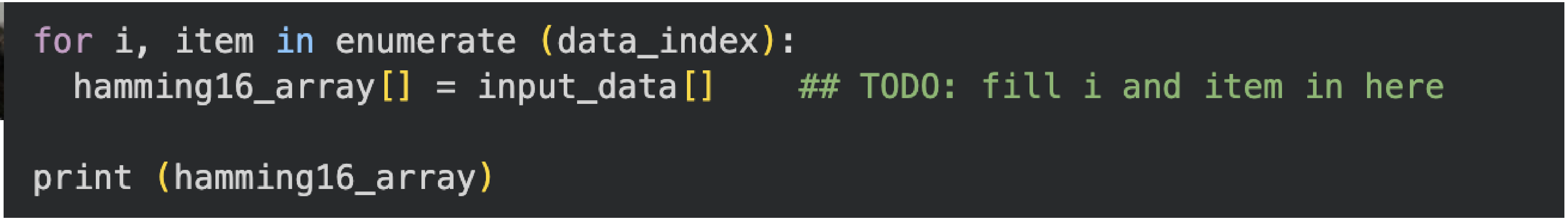
Xiang Li

Spring 2026

COE COLLEGE

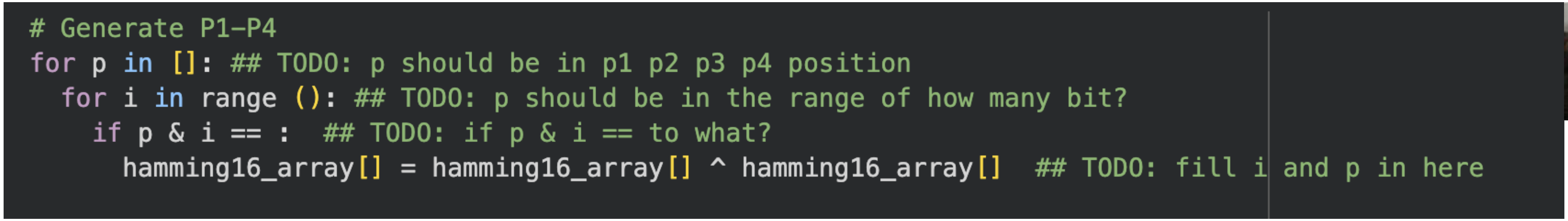
Current roadblocks in Hamming Python:



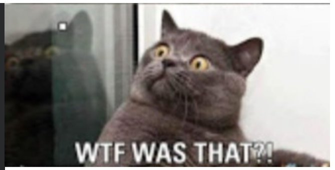


```
for i, item in enumerate (data_index):
  hamming16_array[] = input_data[]    ## TODO: fill i and item in here

print (hamming16_array)
```



```
# Generate P1-P4
for p in []: ## TODO: p should be in p1 p2 p3 p4 position
  for i in range (): ## TODO: p should be in the range of how many bit?
    if p & i == :  ## TODO: if p & i == to what?
      hamming16_array[] = hamming16_array[] ^ hamming16_array[]  ## TODO: fill i and p in here
```



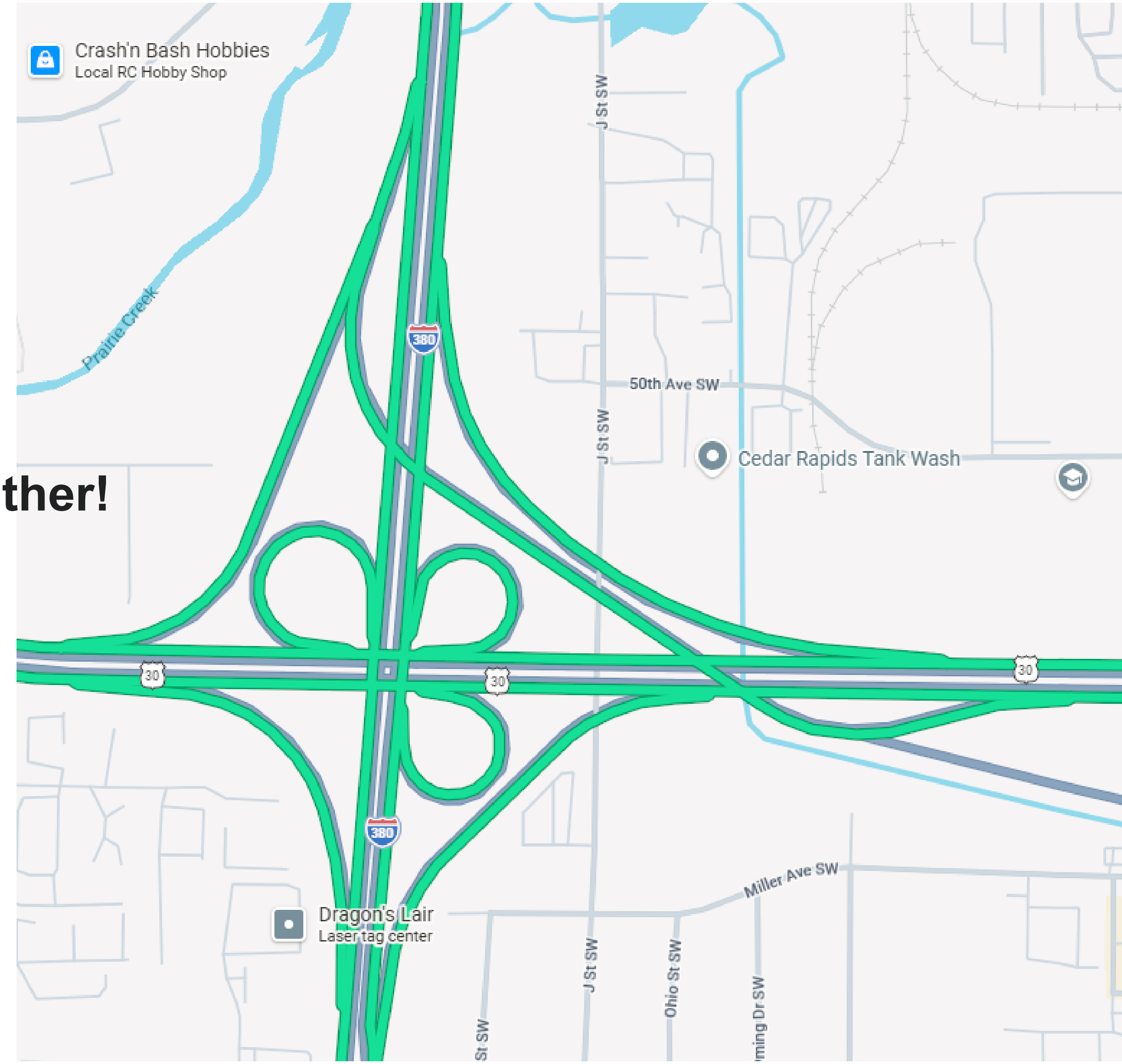
*Cat meme graph is from the internet, somewhere.

We can fix this, by some practice

All computational problem is just:

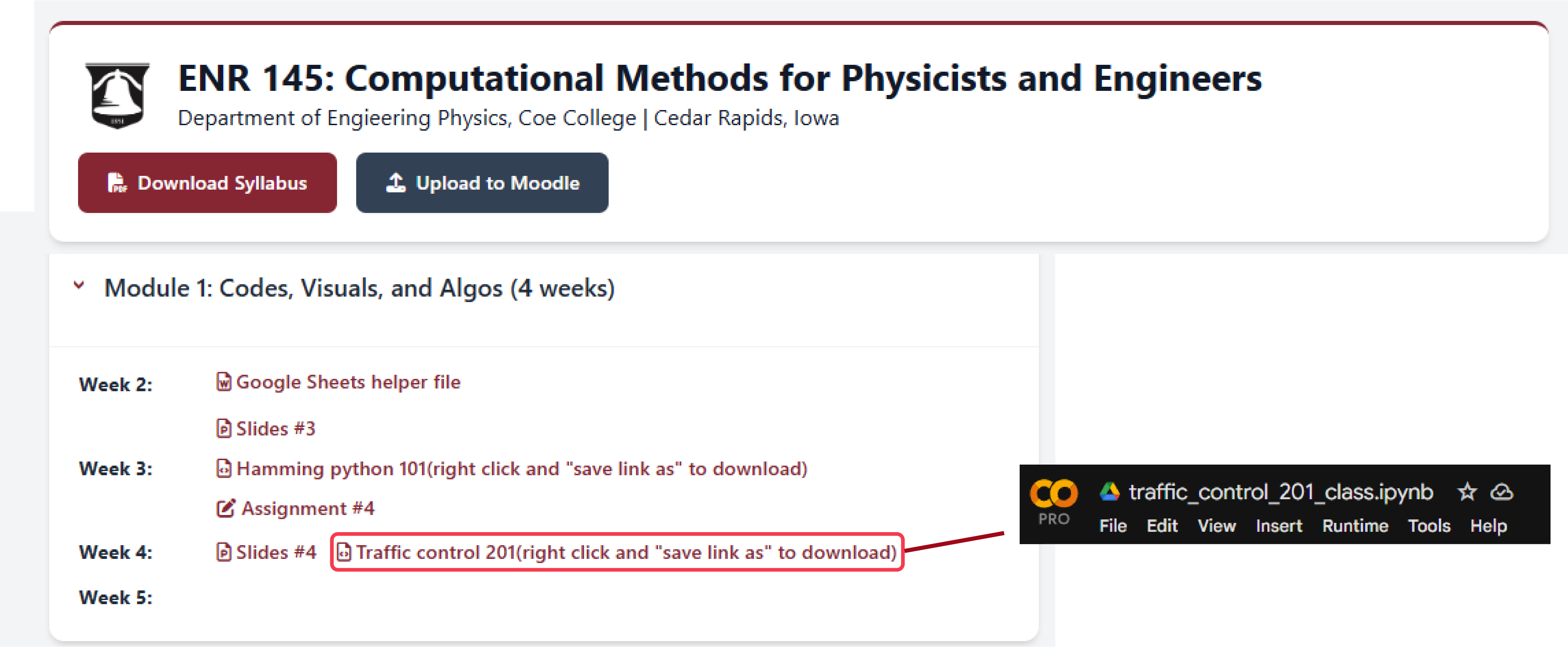
- **Encoding (give number meaning)**
- **Decoding (find meaning from number)**
- **Data manipulation (math out the solution)**

So, let's try to do a traffic control problem together!



*Lane map is found on google map in Cedar Rapids.

How to find the Colab notebook



ENR 145: Computational Methods for Physicists and Engineers

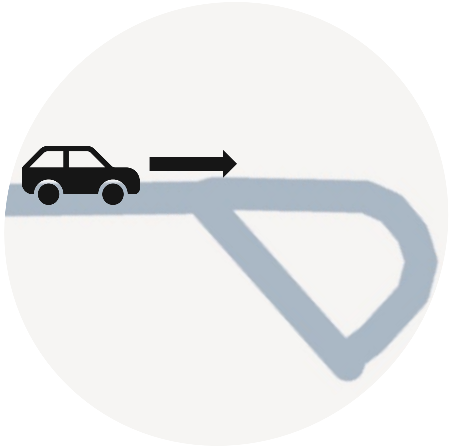
Department of Engieering Physics, Coe College | Cedar Rapids, Iowa

Module 1: Codes, Visuals, and Algos (4 weeks)

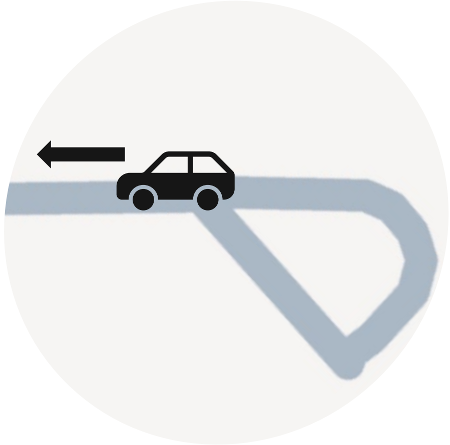
Week 2:	Google Sheets helper file Slides #3
Week 3:	Hamming python 101(right click and "save link as" to download) Assignment #4
Week 4:	Slides #4 Traffic control 201(right click and "save link as" to download)
Week 5:	

Encoding the traffic: level 0

A car is moving in a straight lane with a dead-end.

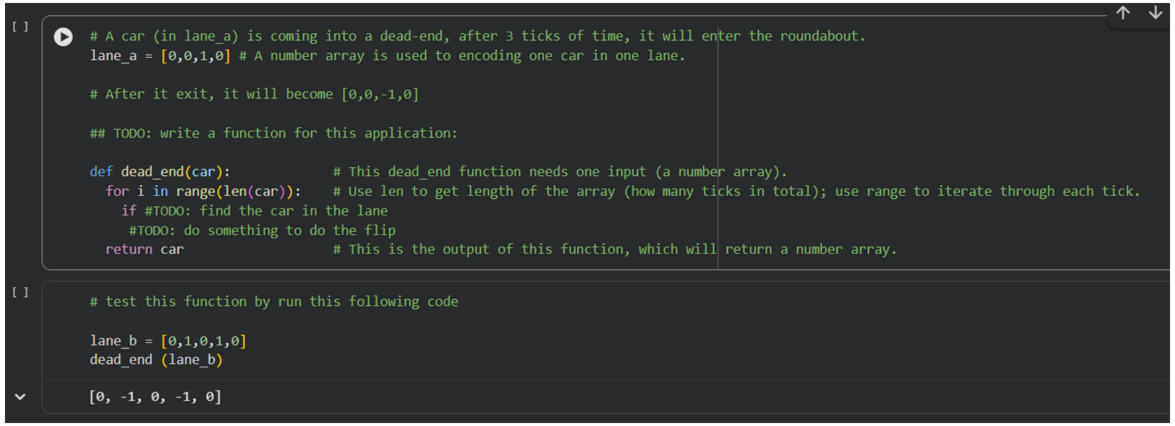


Before: lane_a = [0,0,1,0]



After: lane_a = [0,0,-1,0]

Can you write a function to do this in Python?



```
# A car (in lane_a) is coming into a dead-end, after 3 ticks of time, it will enter the roundabout.
lane_a = [0,0,1,0] # A number array is used to encoding one car in one lane.

# After it exit, it will become [0,0,-1,0]

## TODO: write a function for this application:

def dead_end(car):              # This dead_end function needs one input (a number array).
  for i in range(len(car)):    # Use len to get length of the array (how many ticks in total); use range to iterate through each tick.
    if #TODO: find the car in the lane
     #TODO: do something to do the flip
  return car                    # This is the output of this function, which will return a number array.
```

```
# test this function by run this following code

lane_b = [0,1,0,1,0]
dead_end (lane_b)
```

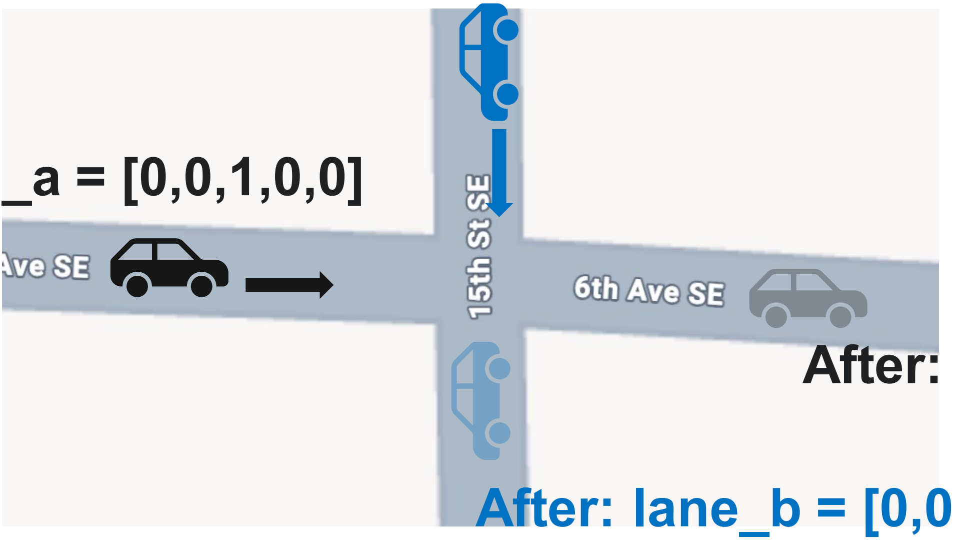
```
[0, -1, 0, -1, 0]
```

*Lane map is found on google map in Cedar Rapids.

Encoding the traffic: level 1

Two cars meet in one intersection.

Before: lane_b = [0,0,1,0,0]



Before: lane_a = [0,0,1,0,0]

After: lane_a = [0,0,1,0,0]

After: lane_b = [0,0,0,1,0]

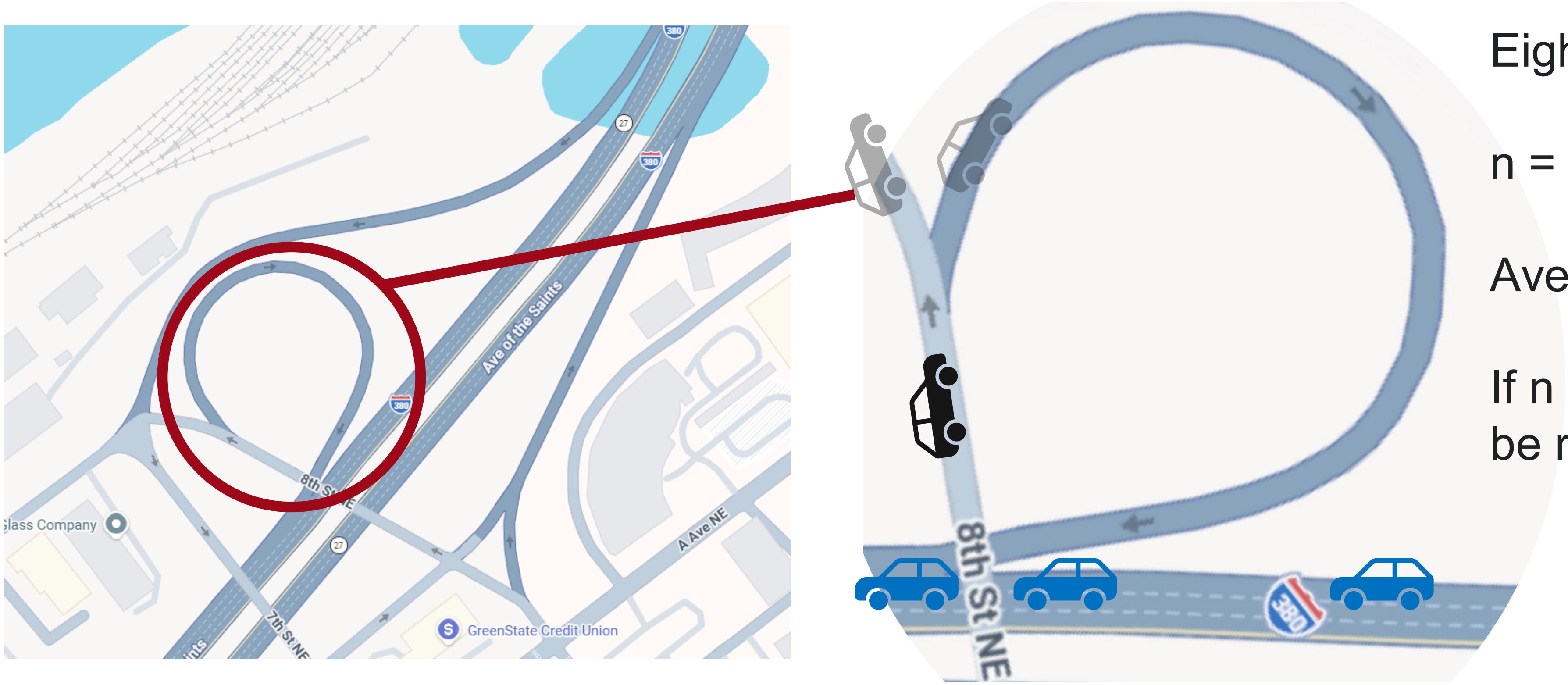
It waited for 1 tick, so lane_a will be clear.

Can you write a function to do this in Python?

*Lane map is found on google map in Cedar Rapids.

Encoding the traffic: level 2

A car in one lane might have different states (left/right, merge/no merge).



Eighth_ST = [0,n,n,0,0]

n = left or right?

Ave_380 = [1,1,1,0,1,1,0,1,1]

If n merge to 380, it needs to be removed from 8th St array.

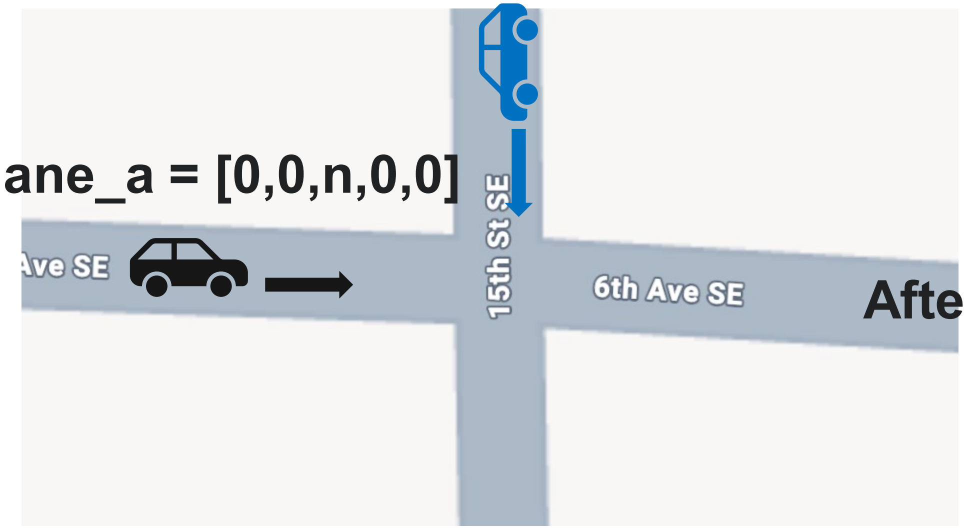
COE COLLEGE

*Lane map is found on google map in Cedar Rapids.

Encoding the traffic: level 2.2

More states, how to check collision?

Before: lane_b = [0,0,n,0,0]



Before: lane_a = [0,0,n,0,0]

After:?

After: ?

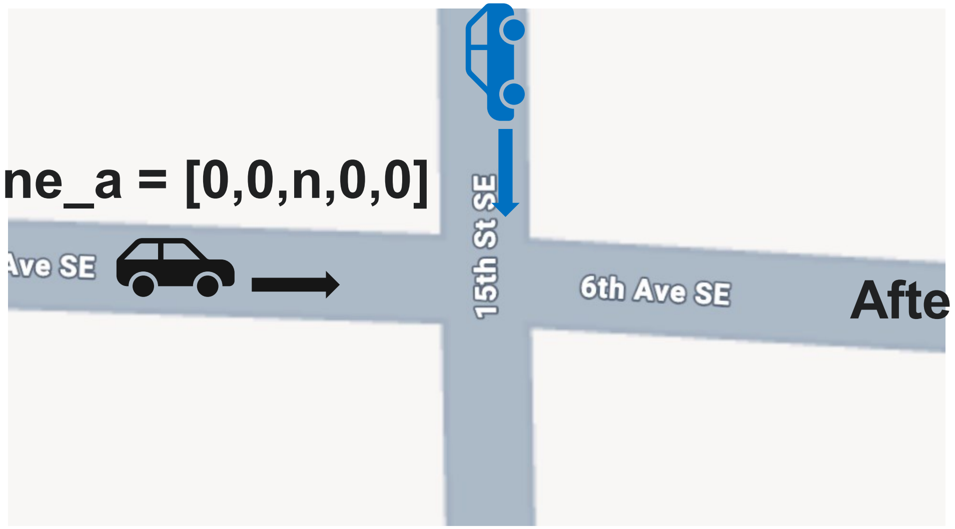
*Lane map is found on google map in Cedar Rapids.

Encoding the traffic: level 2.3

If collision == true, how to code “right of the way”?

Before: lane_b = [0,0,n,0,0]

Before: lane_a = [0,0,n,0,0]



After:?

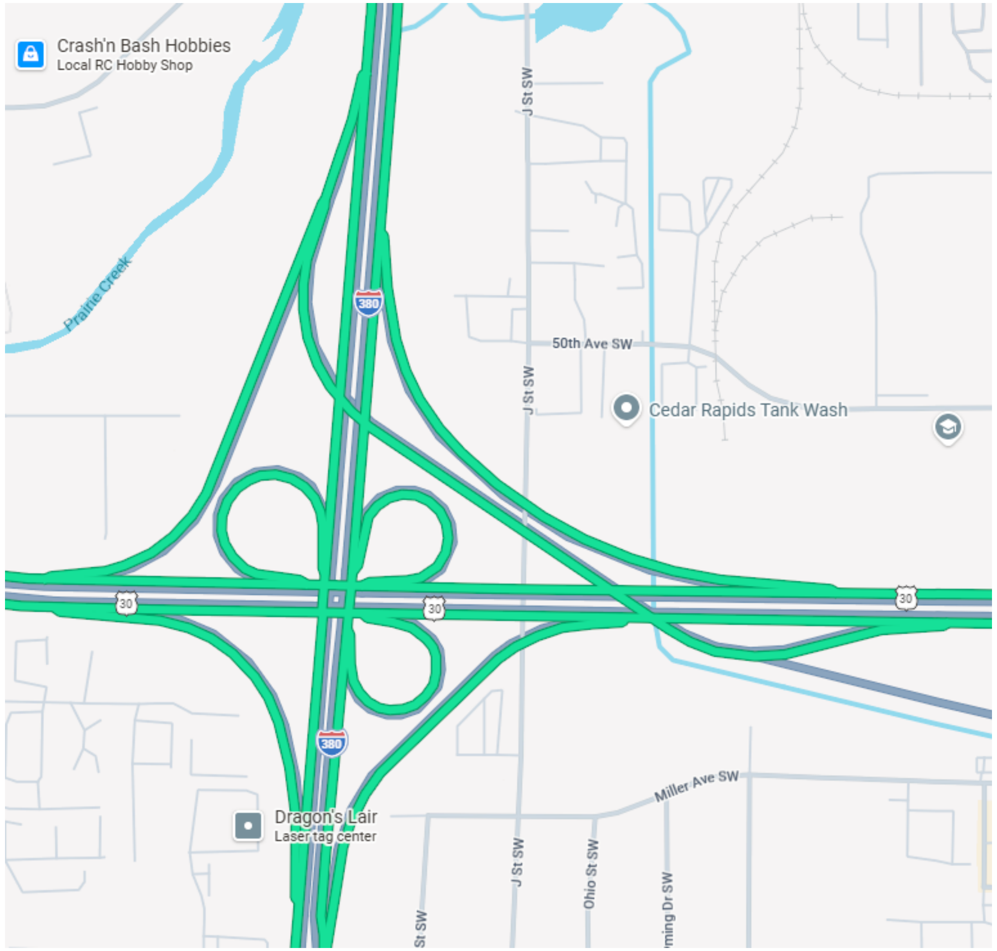
After: ?

COE COLLEGE

*Lane map is found on google map in Cedar Rapids.

Encoding the traffic: level 3

Cloverleaf interchange



How many lanes?
380N, 380S, 30E, 30W

How many options for output?
3 for each lane, how to code?

Bottleneck: need to remove car from the original lane once it's gone.

COE COLLEGE

*Lane map is found on google map in Cedar Rapids.