Lesson 6
Debugging & Testing

Programming Fundamentals in Python

Lesson 5 Recap

- Homework: Sierpinski
- Bonus: Sudoku

Class Materials

github.com/DavidYKay/python-fundamentals

Today's Goal

- Debug and fix someone else's code
- Test and understand a black box

Testing: Caesar Cipher Demo

Debugging: “Google Maps” Demo

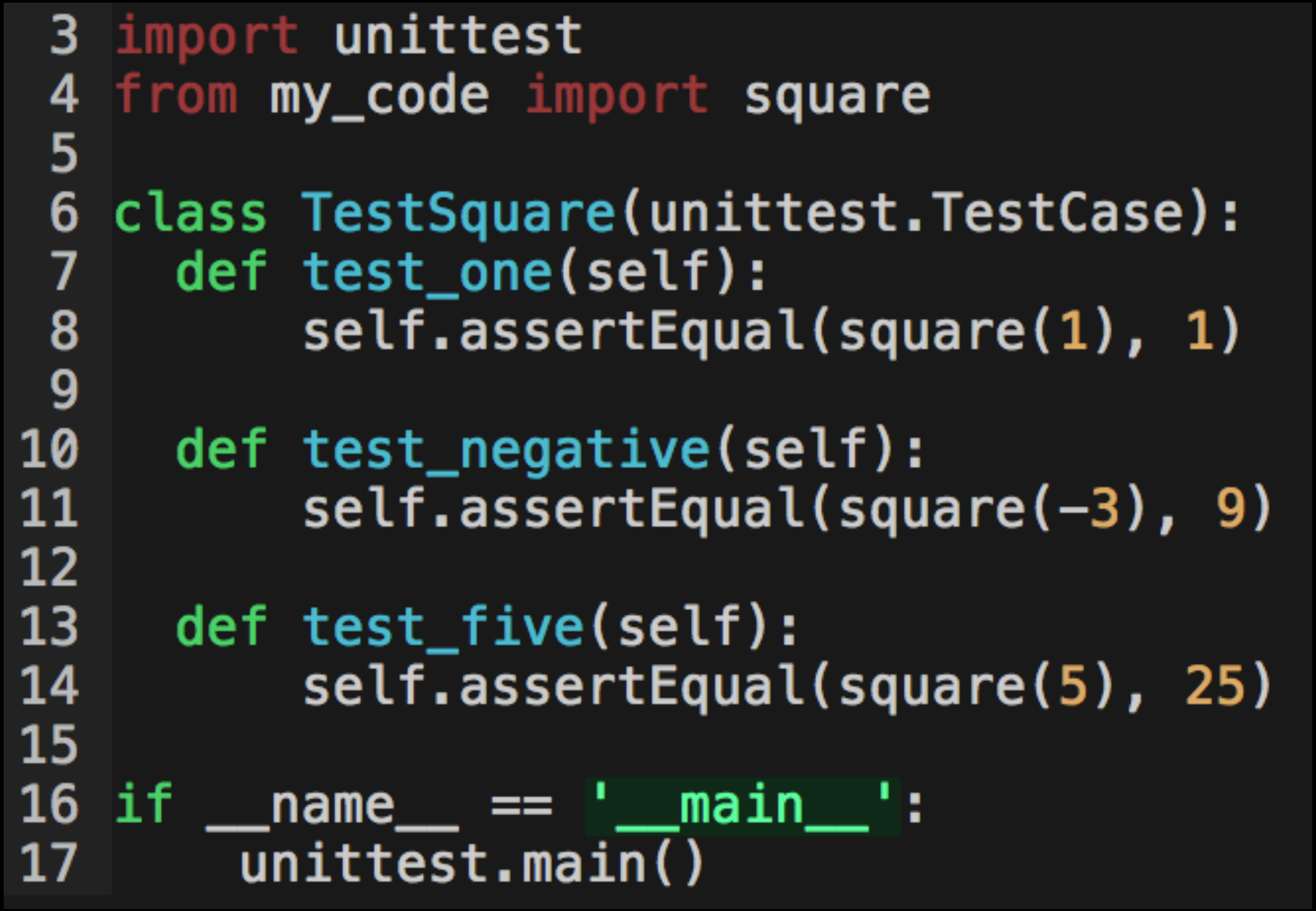
Breakdown

- Testing
- Debugging
- ASCII
- Graph data structure

Testing

iPhone 6 & iPhone 6 Plus All Torture Test

Unit Testing



```python
import unittest
from my_code import square

class TestSquare(unittest.TestCase):
  def test_one(self):
      self.assertEqual(square(1), 1)

  def test_negative(self):
      self.assertEqual(square(-3), 9)

  def test_five(self):
      self.assertEqual(square(5), 25)

if __name__ == '__main__':
    unittest.main()
```

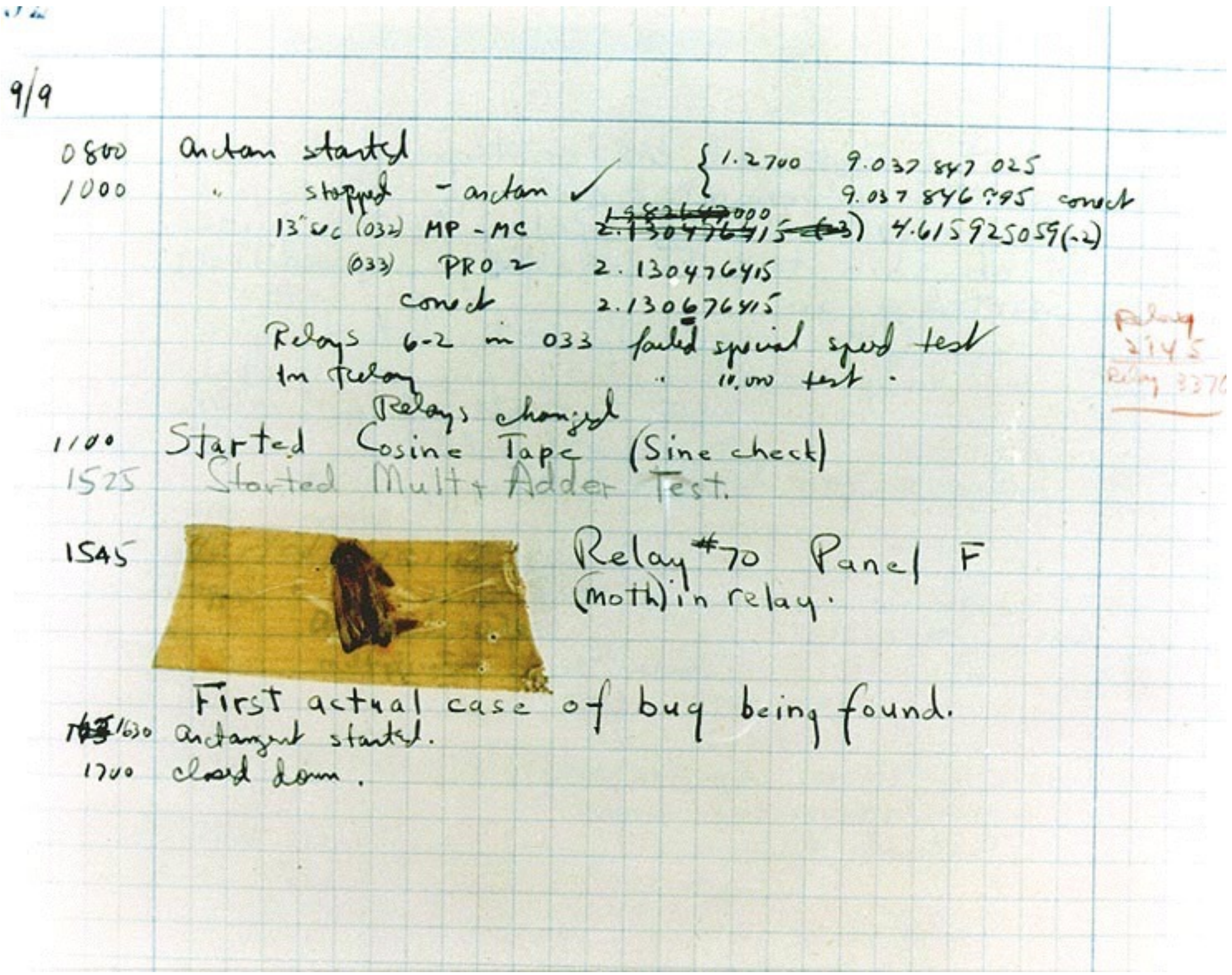
Unit Testing

- Set expectations
- Run the code-under-test
- Does reality meet expectations?

Black Box

- Can’t see what’s inside
- Does it behave how I expect it to?
- Often used for UI testing
- Unit testing often called “white box testing”

Debugging



9/9

0800 Arctan started

1000 " stopped - arctan ✓ { 1.2700 9.037 847 025

9.037 846 995 correct

13" sec (032) MP - MC ~~1.982147000~~ ~~2.130476415~~ ~~(23)~~) 4.615925059(-2)

(033) PRO 2 2.130476415

correct 2.130676415

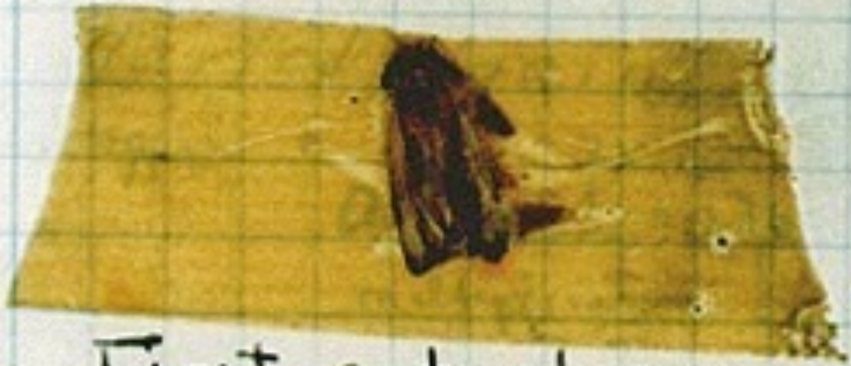
Relays 6-2 in 033 failed special speed test

In relay " 10,000 test .

Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Mult + Adder Test.

1545  Relay #70 Panel F

(moth) in relay.

First actual case of bug being found.

~~1525~~ 1630 arctangent started.

1700 closed down.

Relay 2145

Relay 3370

Debugging

- What is wrong?
- Where is the problem?
- How to fix it?

Debugging Tips

- When programming, change one thing at a time!
- What’s broken? The last thing you touched.

Binary Search

- EXCELLENT tool for fixing bugs
- Bisect until you find the problem

PDB

```
import pdb; pdb.set_trace()
```

PDB Prompt

```
l - list 11 lines around current line

b [N] - set breakpoint at line N

c - continue

p - print
```

see **homework/Pdb_Commands.pdf** for more

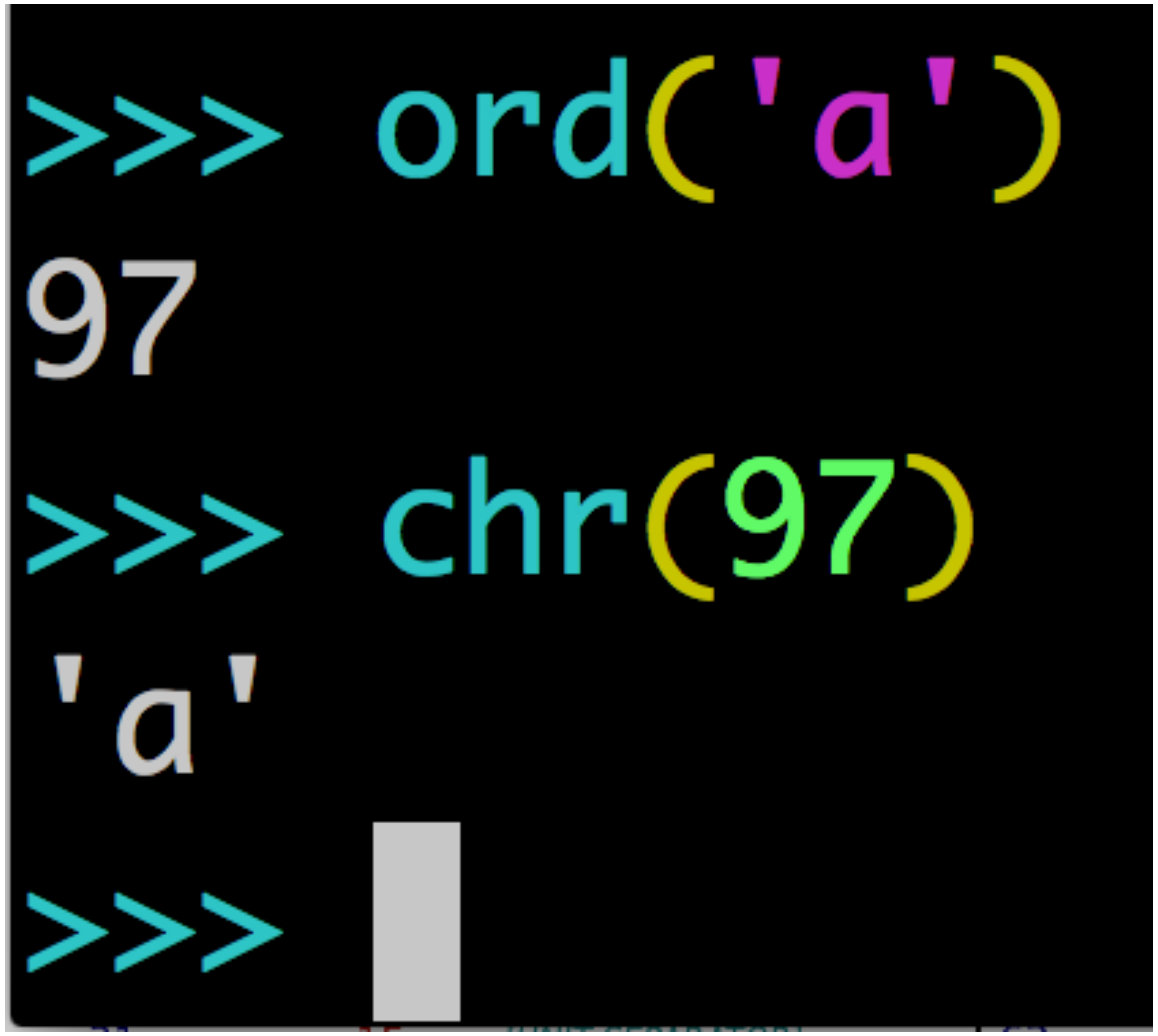
PDB Demo

ASCII

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

ASCII in Python



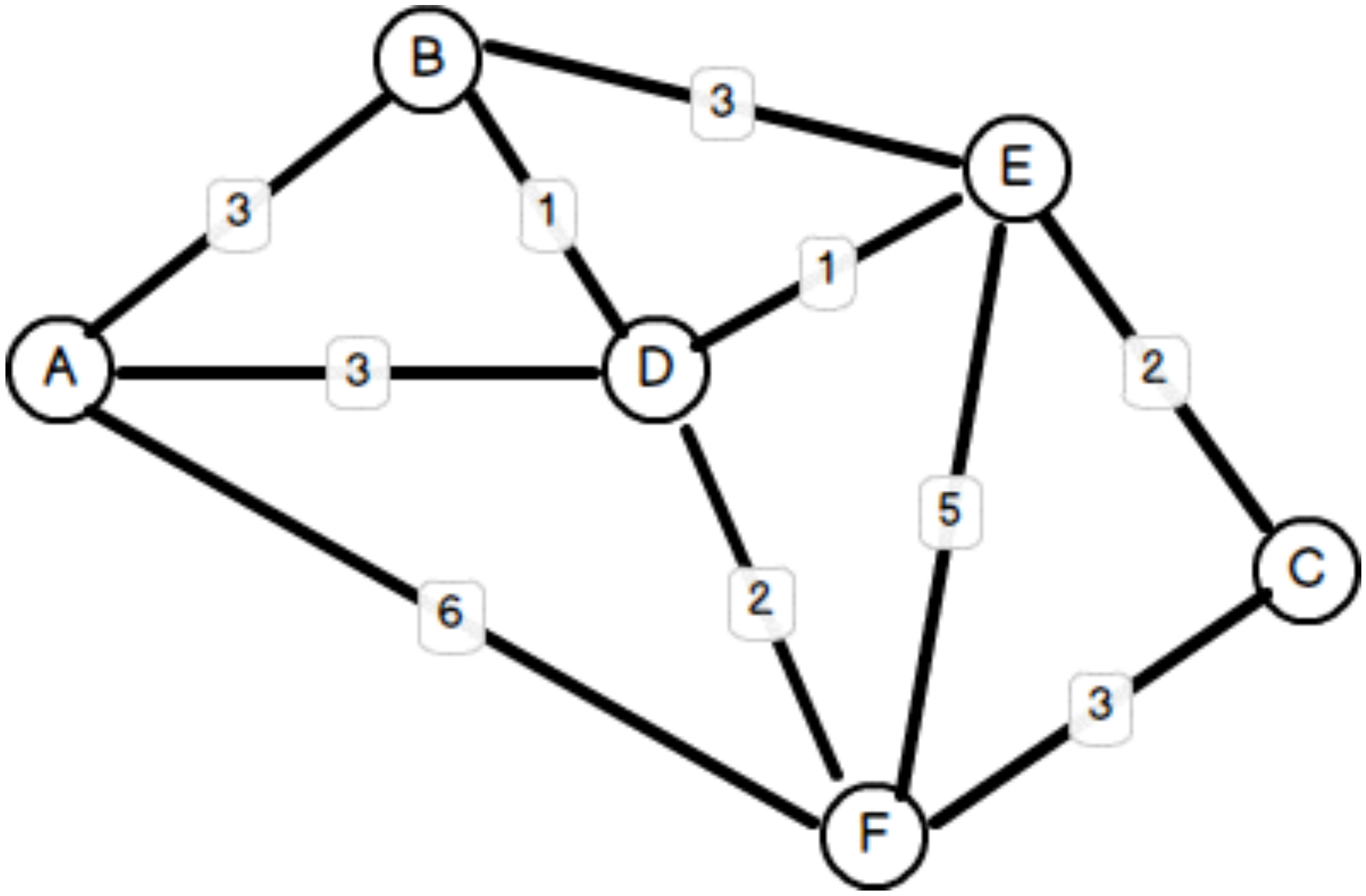
```
>>> ord('a')
97
>>> chr(97)
'a'
>>> 
```

ASCII in Python

- **`ord`** -> convert character to number
- **`chr`** -> convert number to character

Graph

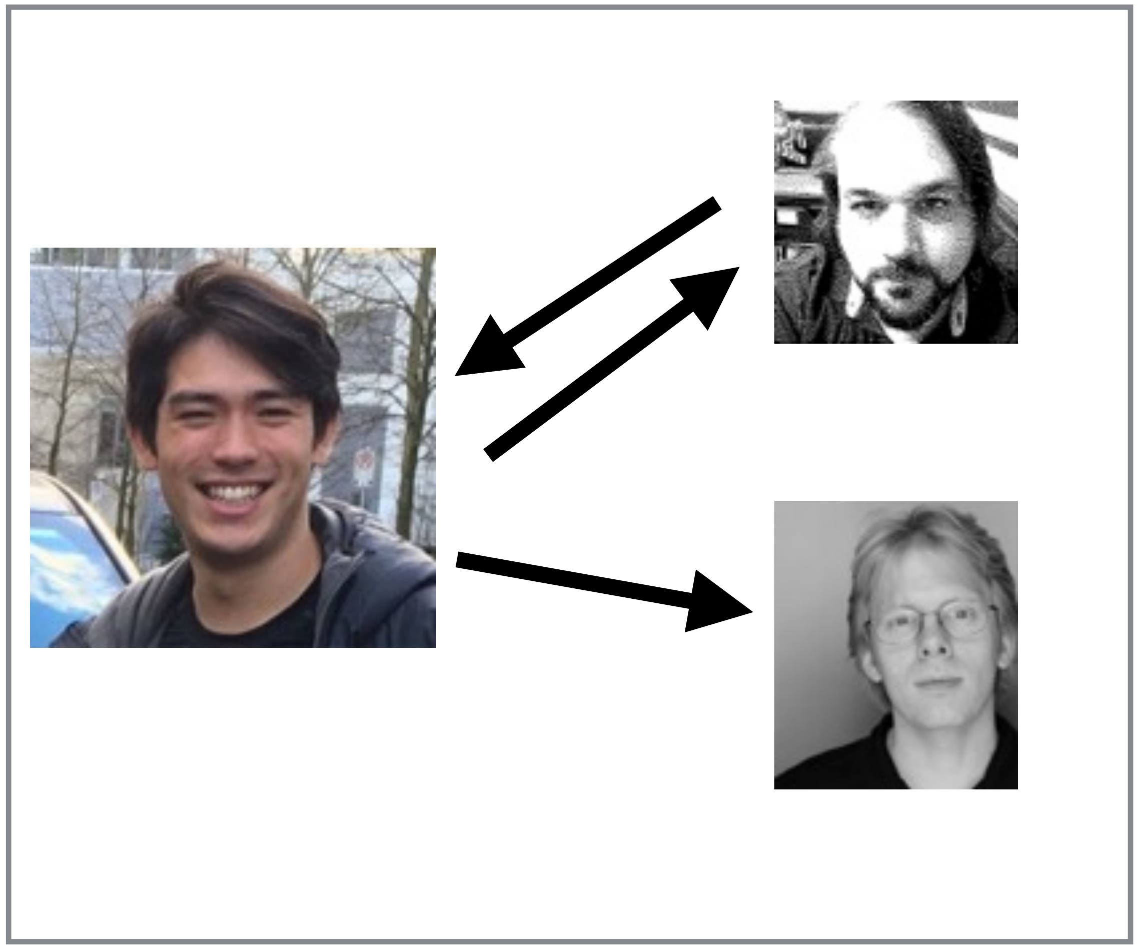
Graph Data Structure

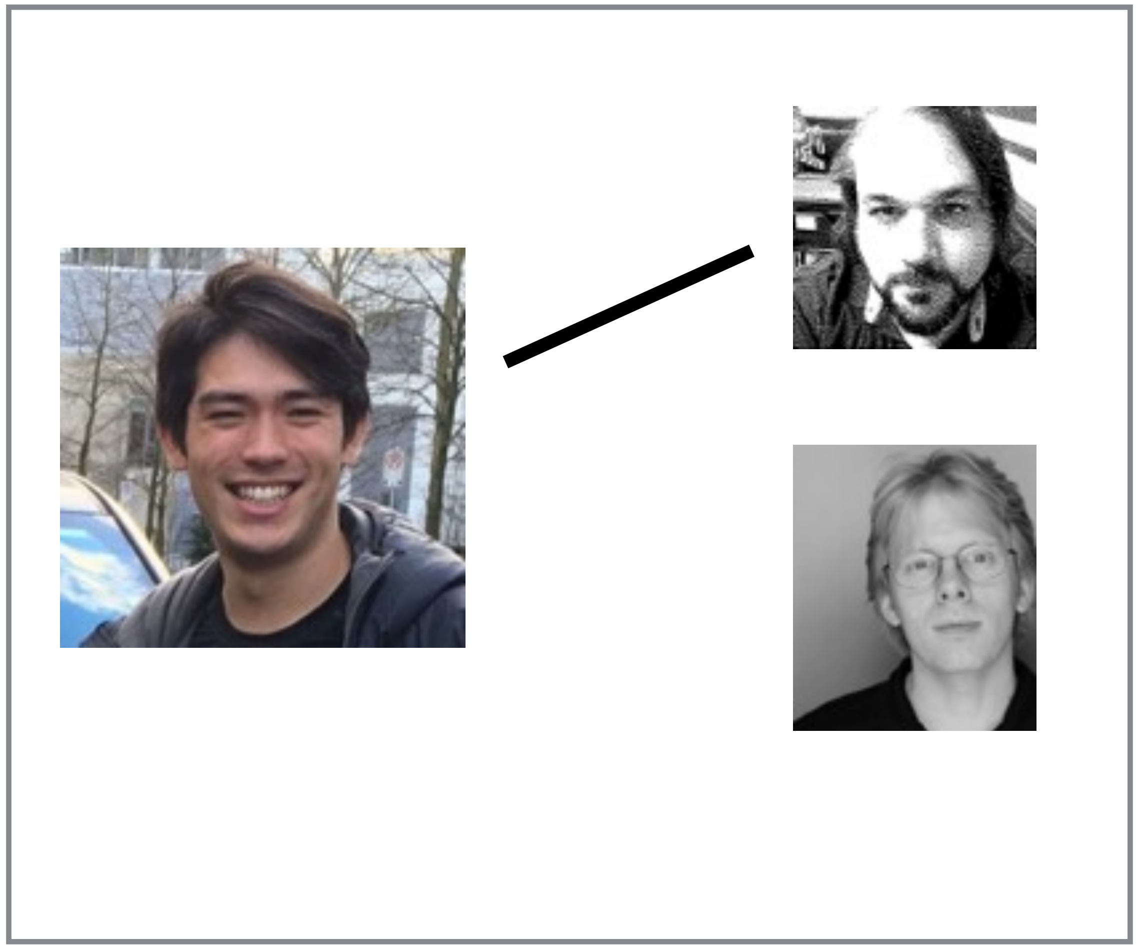


Graph Data Structure

- node - a location in the graph
- edge - a link between two nodes
- weight / cost - a number

Directed / Undirected





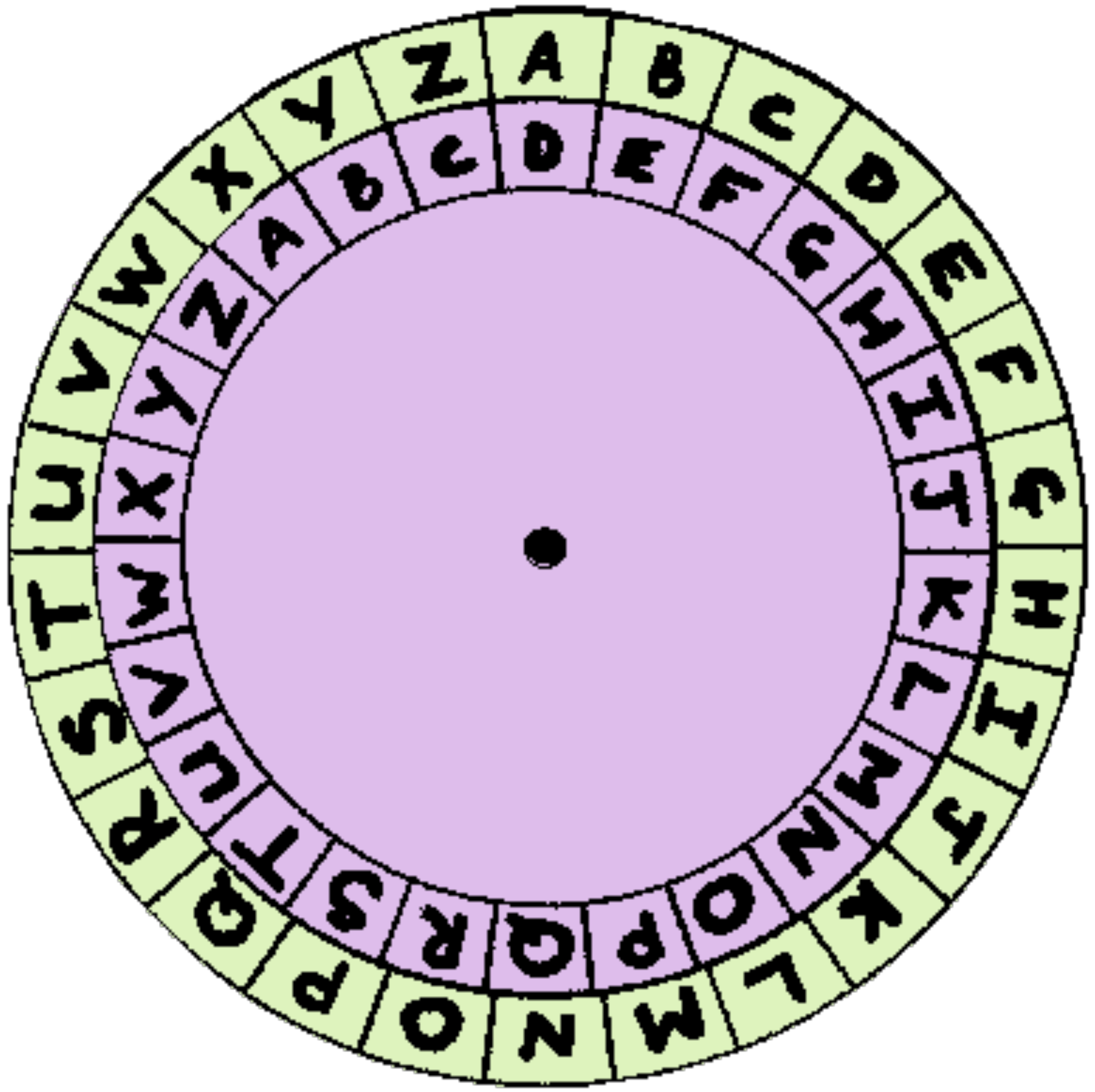
Recap

- Testing
- Debugging
- ASCII
- Graph data structure

Testing Assignment I

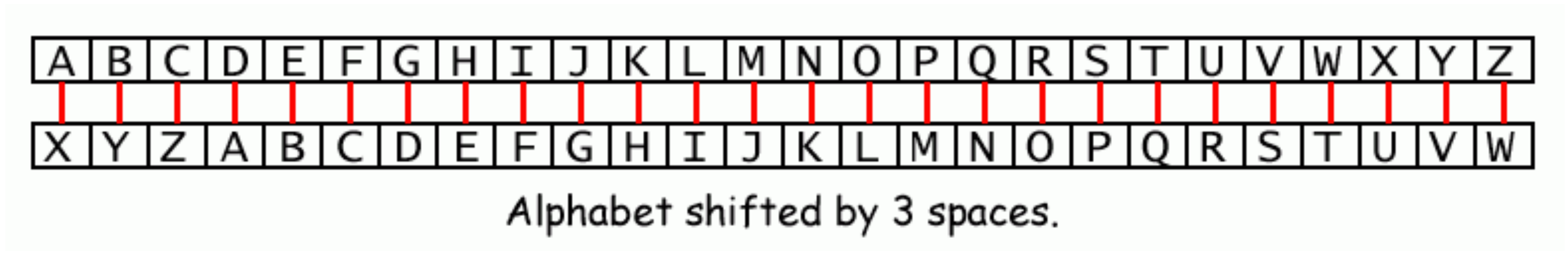
- **caesar.pyc** -> Black box, buggy Caesar Cipher
- Write test cases in **test_caesar.py**
- Determine what inputs cause **caesar.pyc** to fail
    - Email your analysis to me

Caesar Cipher



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
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Caesar Cipher


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
X Y Z A B C D E F G H I J K L M N O P Q R S T U V W

Alphabet shifted by 3 spaces.

hello $\rightarrow$ ebiil

Testing Assignment II

- Given your knowledge from Assignment I, write a working Caesar Cipher and decrypt **encrypted.txt**
  - Hint: offset is a negative number
- Email it to me

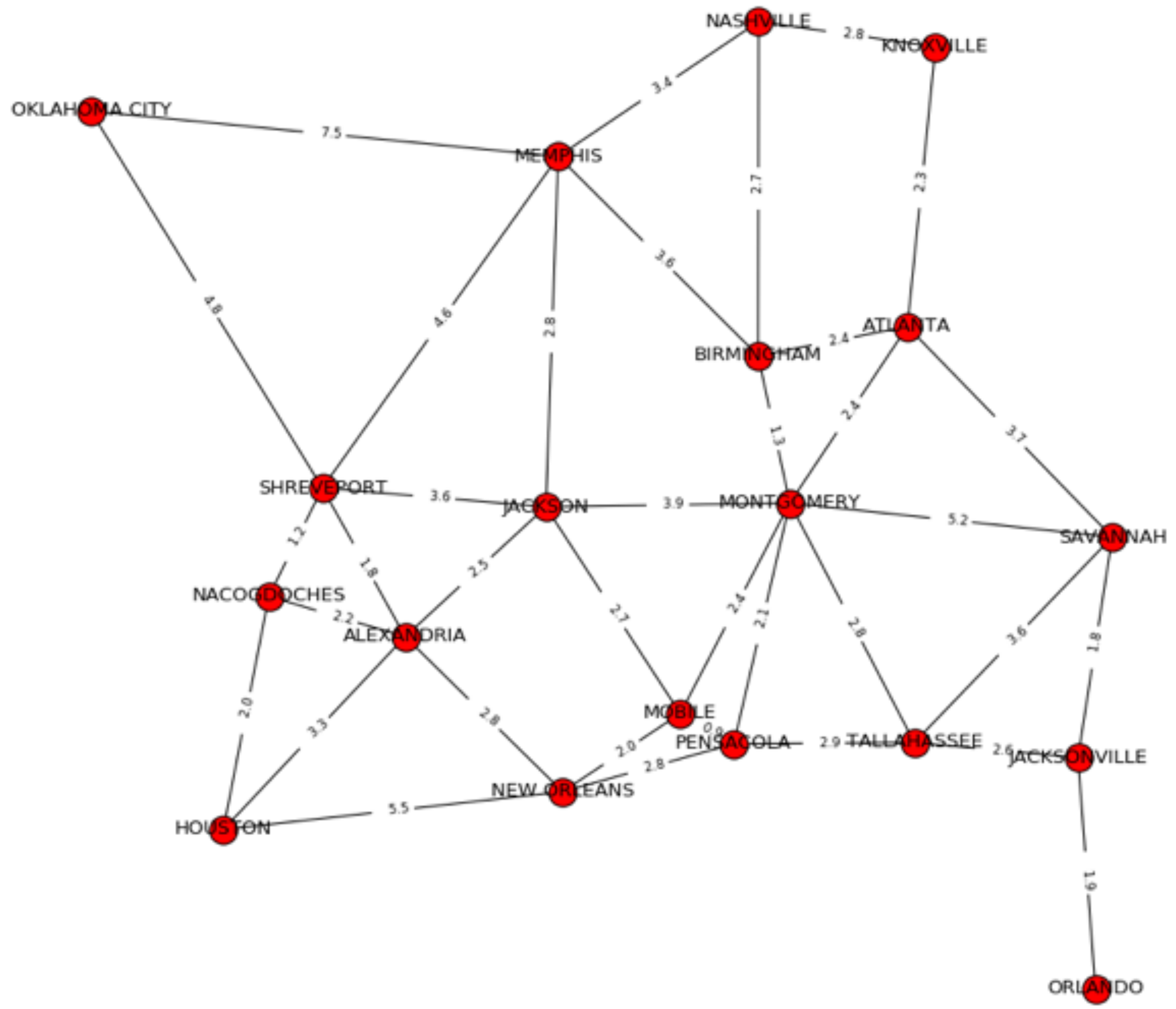
Debugging Assignment

- I have given you a buggy Google Maps clone
    - Should return the fastest route between two cities
- Figure out what's wrong with it
- Fix it

Google Maps

- `pip install matplotlib`
- `pip install networkx`

Google Maps


NASHVILLE
KNOXVILLE
OKLAHOMA CITY
MEMPHIS
ATLANTA
BIRMINGHAM
SHREVEPORT
JACKSON
MONTGOMERY
SAVANNAH
NACOGDOCHES
ALEXANDRIA
MOBILE
PENSACOLA
TALLAHASSEE
JACKSONVILLE
NEW ORLEANS
HOUSTON
ORLANDO
2.8
3.4
7.5
2.7
2.3
3.6
4.8
4.6
2.8
2.4
1.3
2.4
3.7
3.6
3.9
5.2
1.2
1.8
2.5
2.7
2.4
2.1
2.8
3.6
1.8
2.2
2.0
3.3
2.8
0.9
2.0
2.8
2.9
2.6
5.5
1.9

Recap

- Testing
- Debugging
- ASCII
- Graph data structure

Next Week

Grand Finale: Networked Chat!