

# Python Performance

## Investigation by Example

---

@jiffyclub

2018-05-12

# About Me

- Matt Davis aka [@jiffyclub](#)
- Data Engineer at Clover Health (visit us at the Job Fair!)
- Blog: [penandpants.com](#)
- Photos/tweets welcome
- [github.com/jiffyclub/pycon-2018-talk](#)
- License: CC BY-SA 4.0

# Caveats about “Performance”

---



<http://www.attorneysforanimals.org/wp-content/uploads/2016/07/animal-law-book-cat.jpg>



# The Recipe

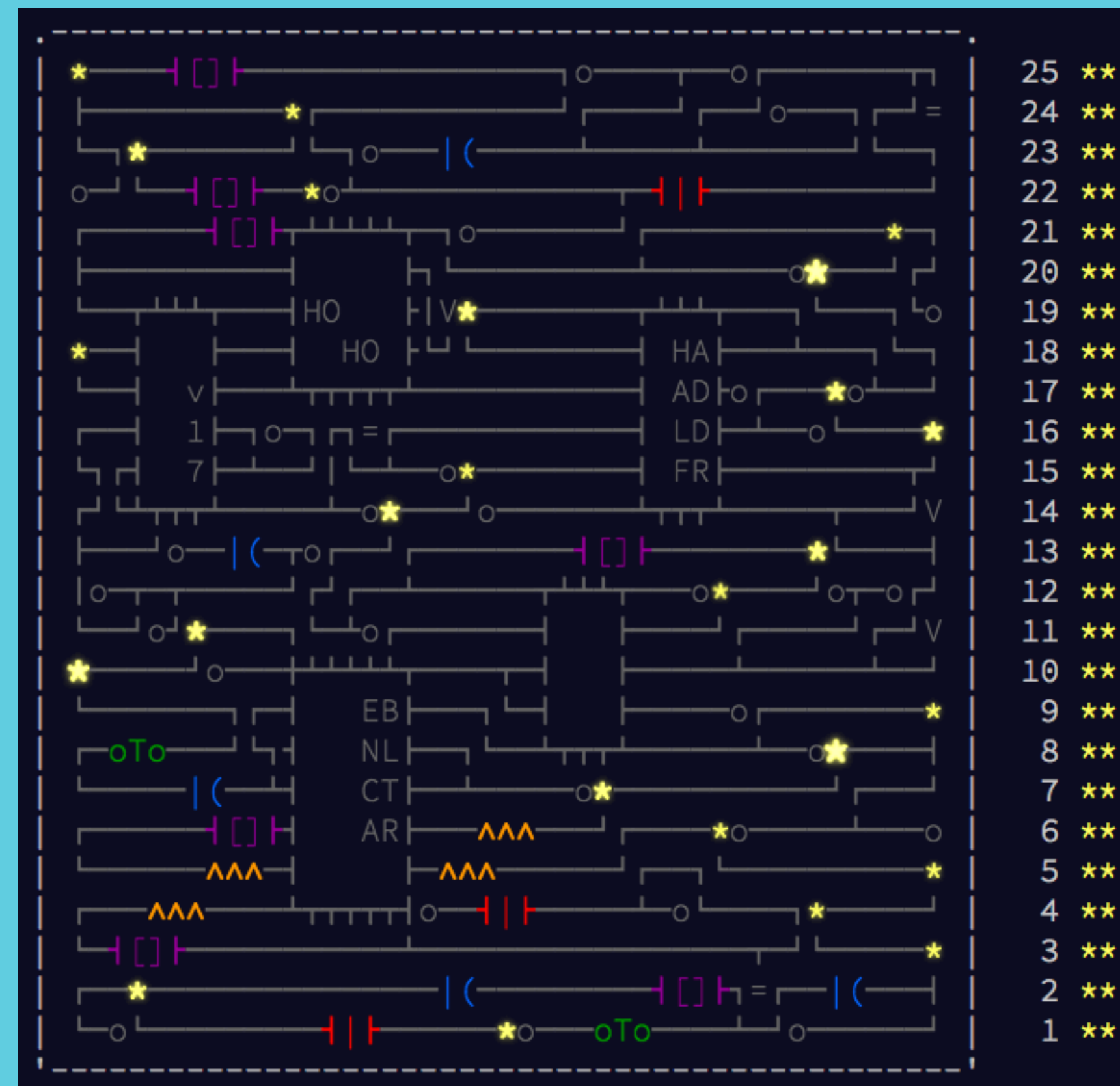
---

- 1. Collect Data**
- 2. Analyze**
- 3. Experiment**
- 4. GOTO 1**



<https://www.etsy.com/listing/257563486/science-cat-funny-pinback-button-cute>

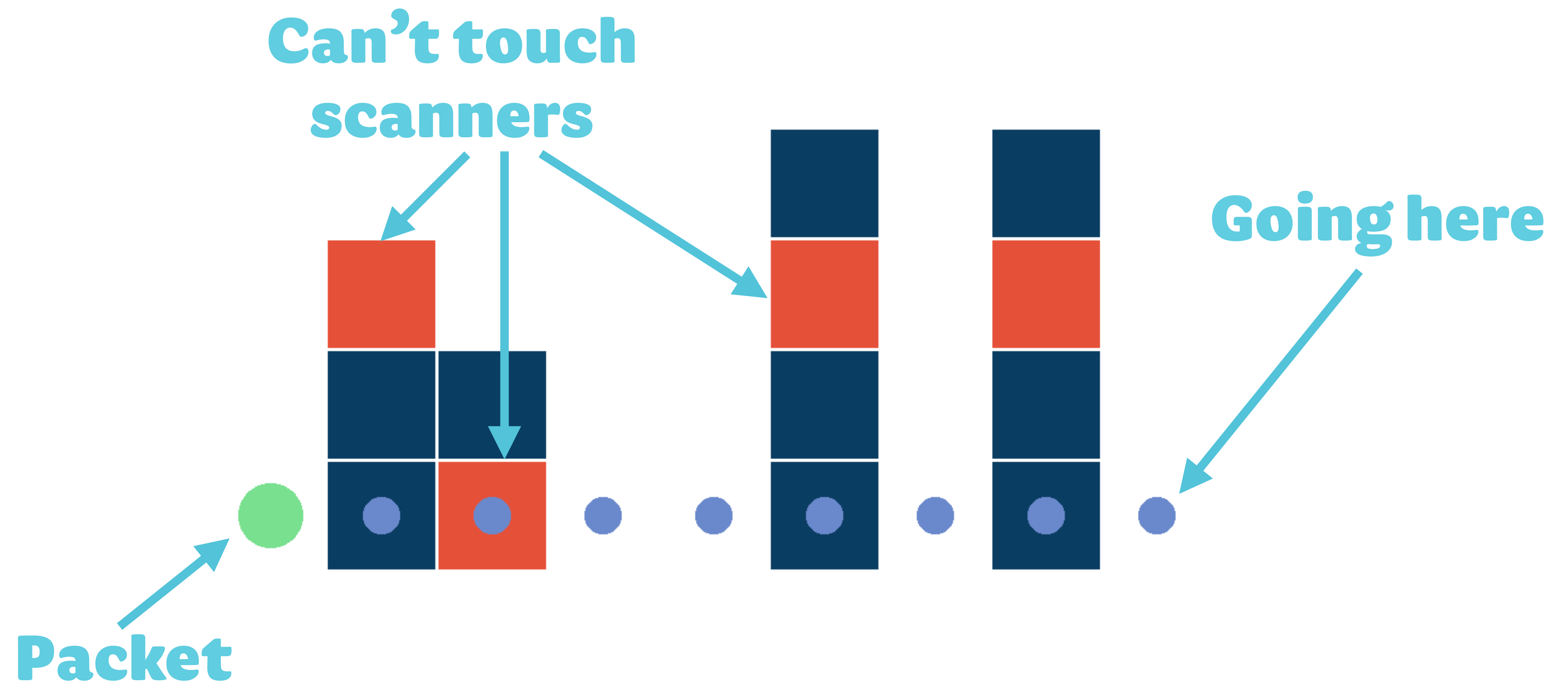
# Backstory



[adventofcode.com/2017](https://adventofcode.com/2017)

# The Assignment: Evade a “Firewall”

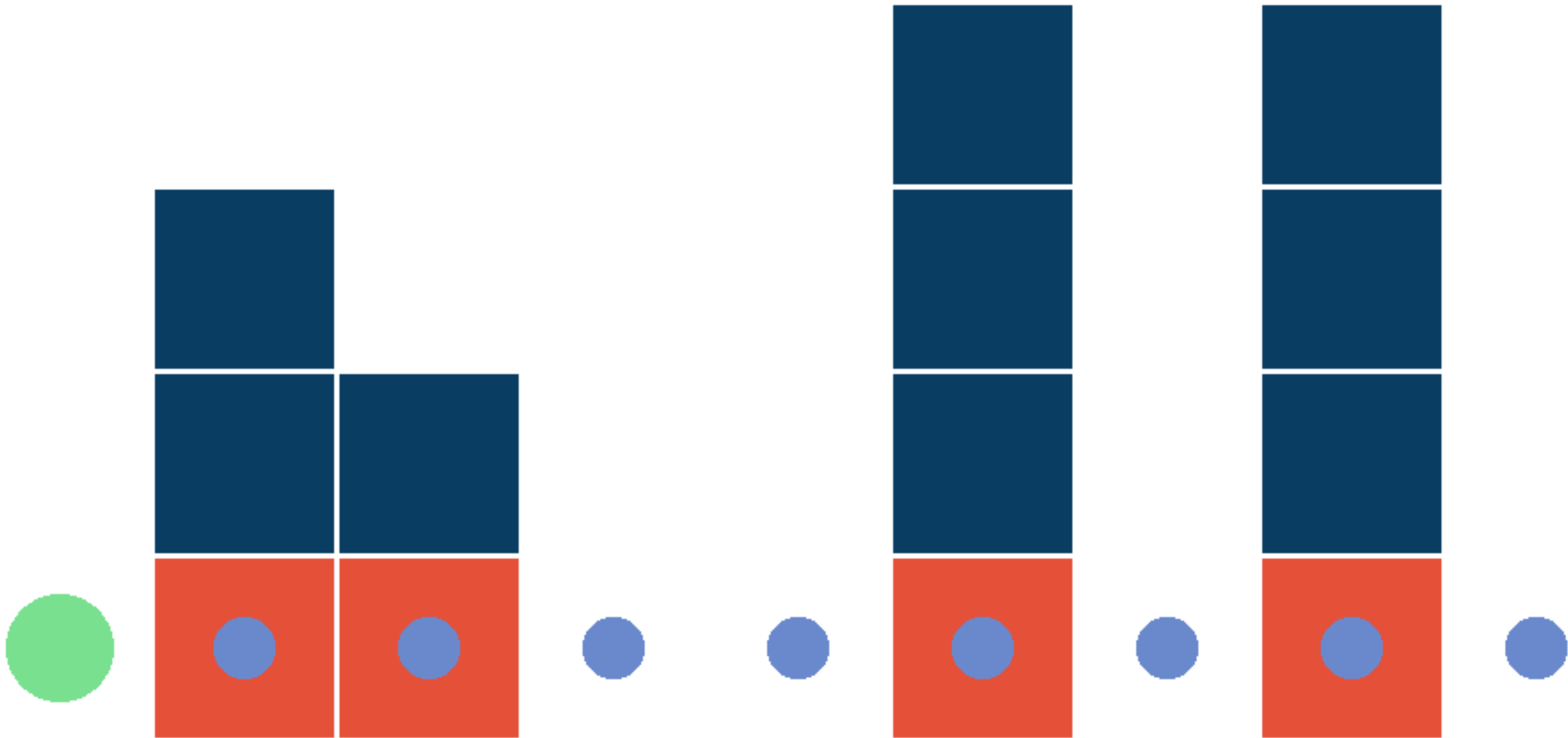
---



# The Assignment: Evade a “Firewall”

---

0



# Solution Outline

---

```
def find_start(firewall) -> int:
```

```
    ...
```

```
    for t_start in itertools.count(0):
```

```
        ...
```

```
        if caught_crossing(firewall, width):
```

```
            ...
```

Did we  
get caught?

```
        else:
```

```
            break
```

```
    return t_start
```

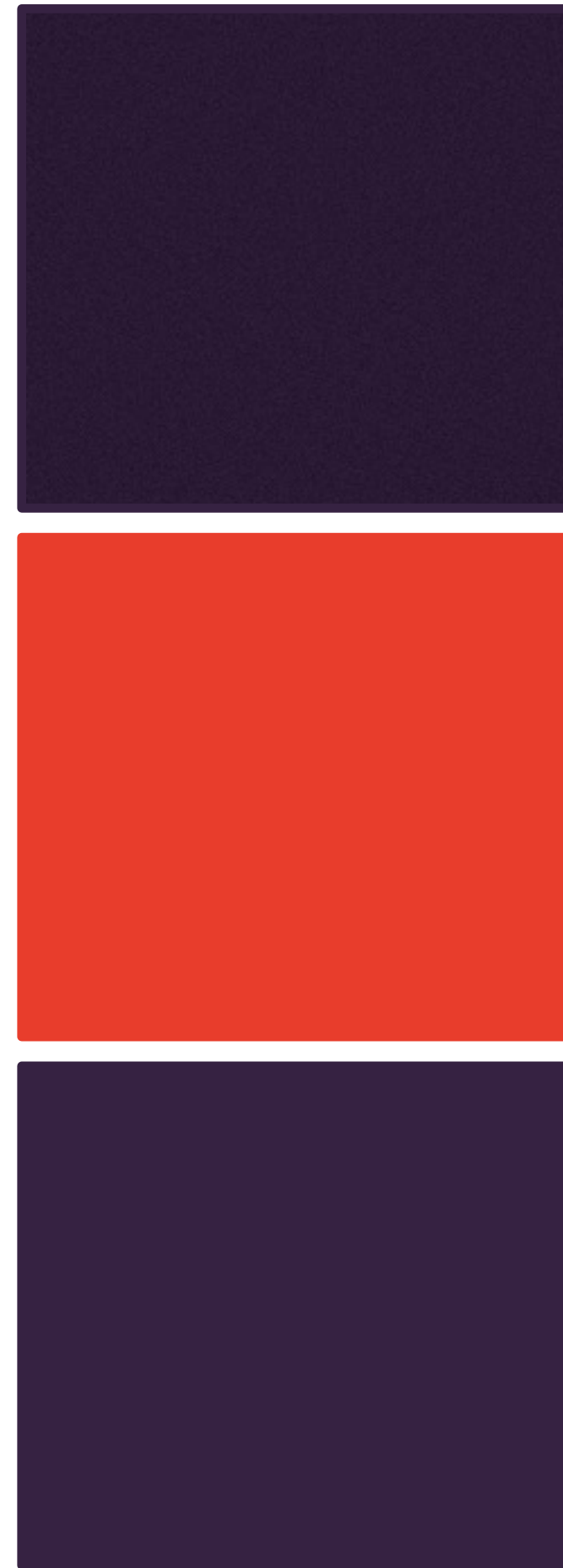
Break out if we didn't get caught



## First Try (Scanner)

---

**class Scanner:**



- height
- position
- direction
- advance()
- copy()

## First Try (firewall)

---

```
firewall = {scanner_slot: Scanner() }
```

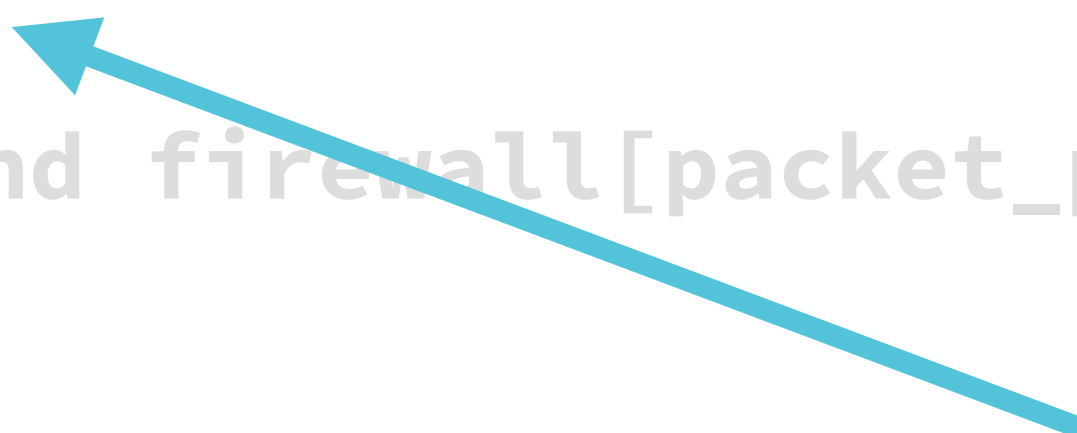
# First Try (caught\_crossing)

---

```
def caught_crossing(
    firewall: Dict[int, Scanner], width: int) -> bool:
    for packet_pos in range(width):
        if packet_pos in firewall and firewall[packet_pos].pos == 0:
            return True

        for scanner in firewall.values():
            scanner.advance()

    return False
```



**Each position in  
firewall**

# First Try (caught\_crossing)

---

```
def caught_crossing(
    firewall: Dict[int, Scanner], width: int) -> bool:
    for packet_pos in range(width):
        if packet_pos in firewall and firewall[packet_pos].pos == 0:
            return True
        for scanner in firewall.values():
            scanner.advance()

    return False
```

**Is there a scanner?**



**Is the scanner at the bottom?**



# First Try (caught\_crossing)

---

```
def caught_crossing(
    firewall: Dict[int, Scanner], width: int) -> bool:
    for packet_pos in range(width):
        if packet_pos in firewall and firewall[packet_pos].pos == 0:
            return True
```

```
    for scanner in firewall.values():
        scanner.advance()
```

```
    return False
```

**Advance scanner  
state one time-step**





# First Try (find\_start)

---

```
def find_start(firewall: Dict[int, Scanner]) -> int:
```

```
    loop_firewall = copy_firewall(firewall)
```

```
    width = max(firewall.keys()) + 1
```

```
    for t_start in itertools.count(0):
```

```
        pre_check_firewall = copy_firewall(loop_firewall)
```

```
        if caught_crossing(loop_firewall, width):
```

```
            loop_firewall = copy_firewall(pre_check_firewall)
```

```
            for scanner in loop_firewall.values():
```

```
                scanner.advance()
```

```
        else:
```

```
            break
```

```
    return t_start
```

**Copy before check**



**Copy after check**



**Did It Work?**

**Answer: 3,823,370 time-steps**

**How Long Did it Take?**

**Did It Work?**

**Answer: 3,823,370 time-steps**

**How Long Did it Take?**

**10 minutes (600 seconds)**



# Thoughtful Time

---



<https://hoschie.deviantart.com/art/Thinking-cat-346209983>

# Collecting Data a.k.a. Profiling

---

## cProfile

Built into Python

Tracks time spent in functions

## Others

line\_profiler

pyflame

plop

nylas-perftools



## Running cProfile

---

```
python -m cProfile \
```

← **Command line**

```
-o output_file \
```

```
my_script.py
```

```
%%prun -q -D output_file
```

**IPython magic**

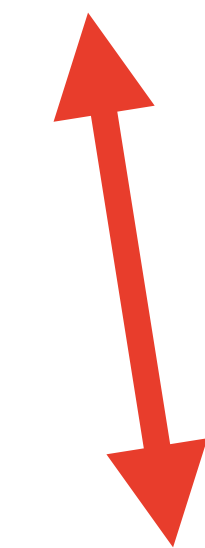
# Viewing Data (pstats)

---

**Loading profile data**

```
In [2]: data = pstats.Stats('slow_mode.cprof')
```

```
In [3]: data.sort_stats('cumulative').print_stats('slow_mode', 2)
```




ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
1	0.000	0.000	808.171	808.171	slow_mode.py:1(<module>)
1	93.039	93.039	808.156	808.156	slow_mode.py:74(find_start)

...

# Viewing Data (pstats)

---

```
In [4]: data.sort_stats('tottime').print_stats(2)
```



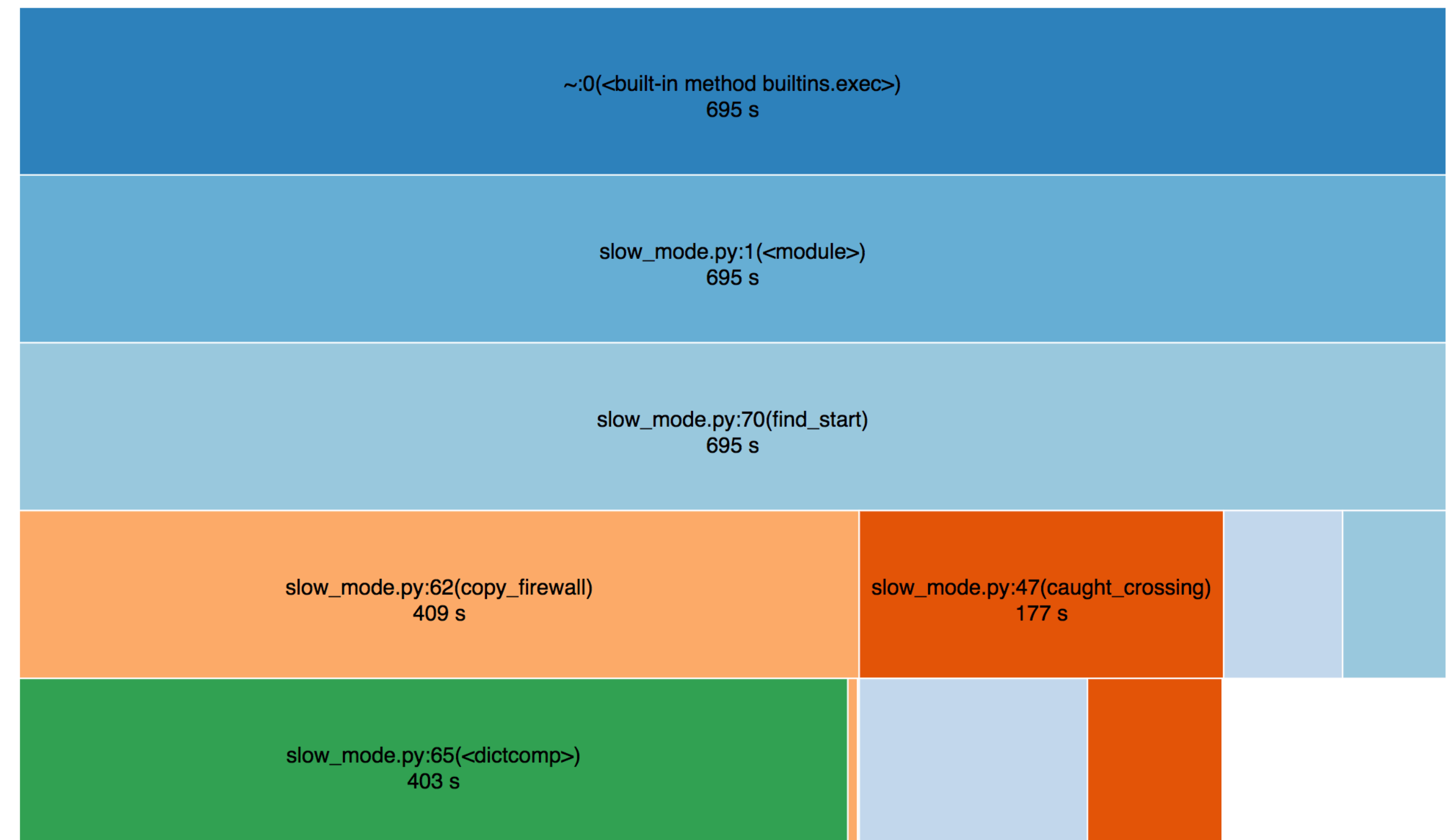
...

ncalls	tottime	percall	cumtime	percall	filename:lineno(function)
328809906	234.089	0.000	364.089	0.000	slow_mode.py:28(copy)
488635144	168.532	0.000	168.532	0.000	slow_mode.py:13(advance)

# SnakeViz

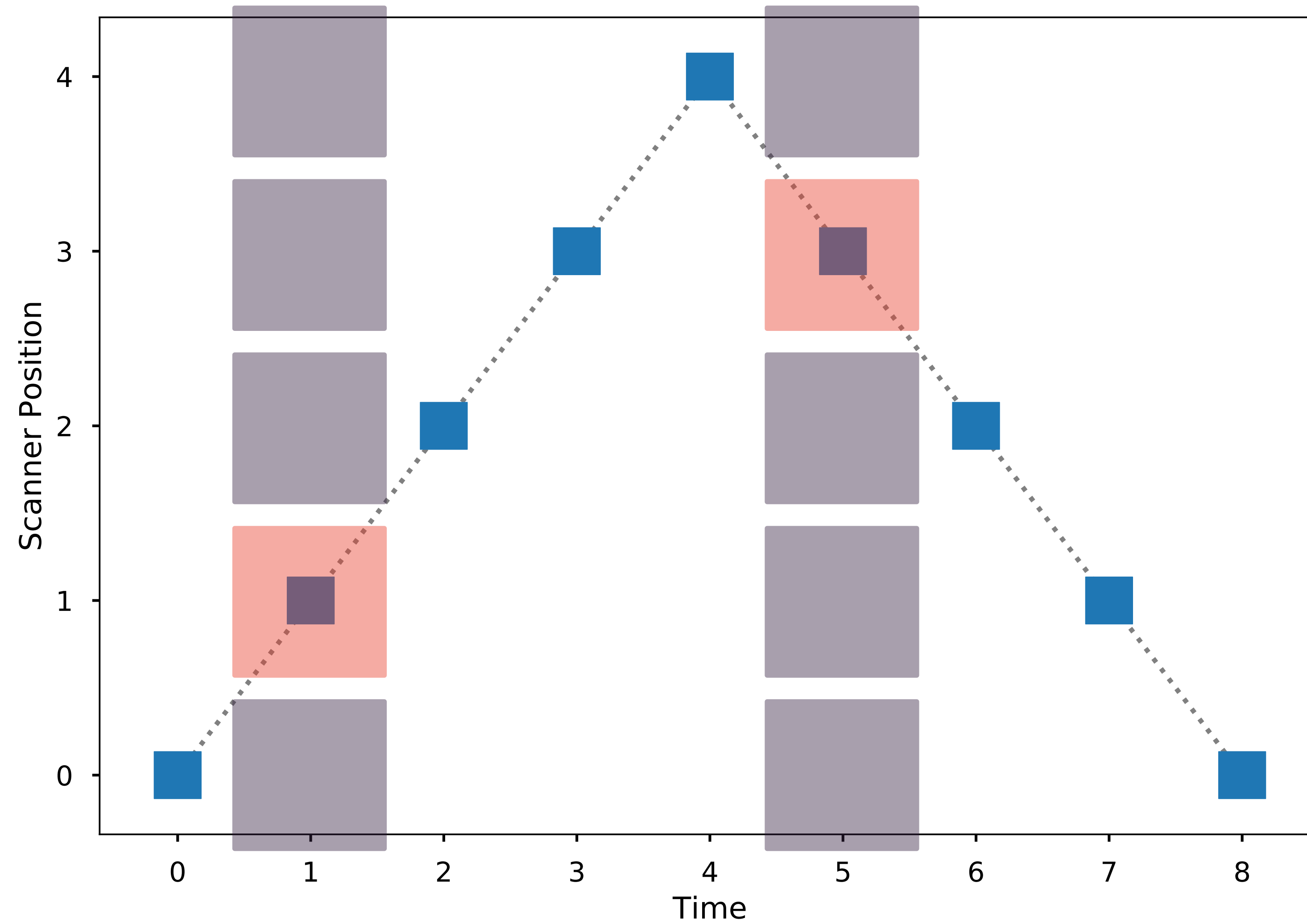
---

```
pip install snakeviz
snakeviz slow_mode.cprof
```



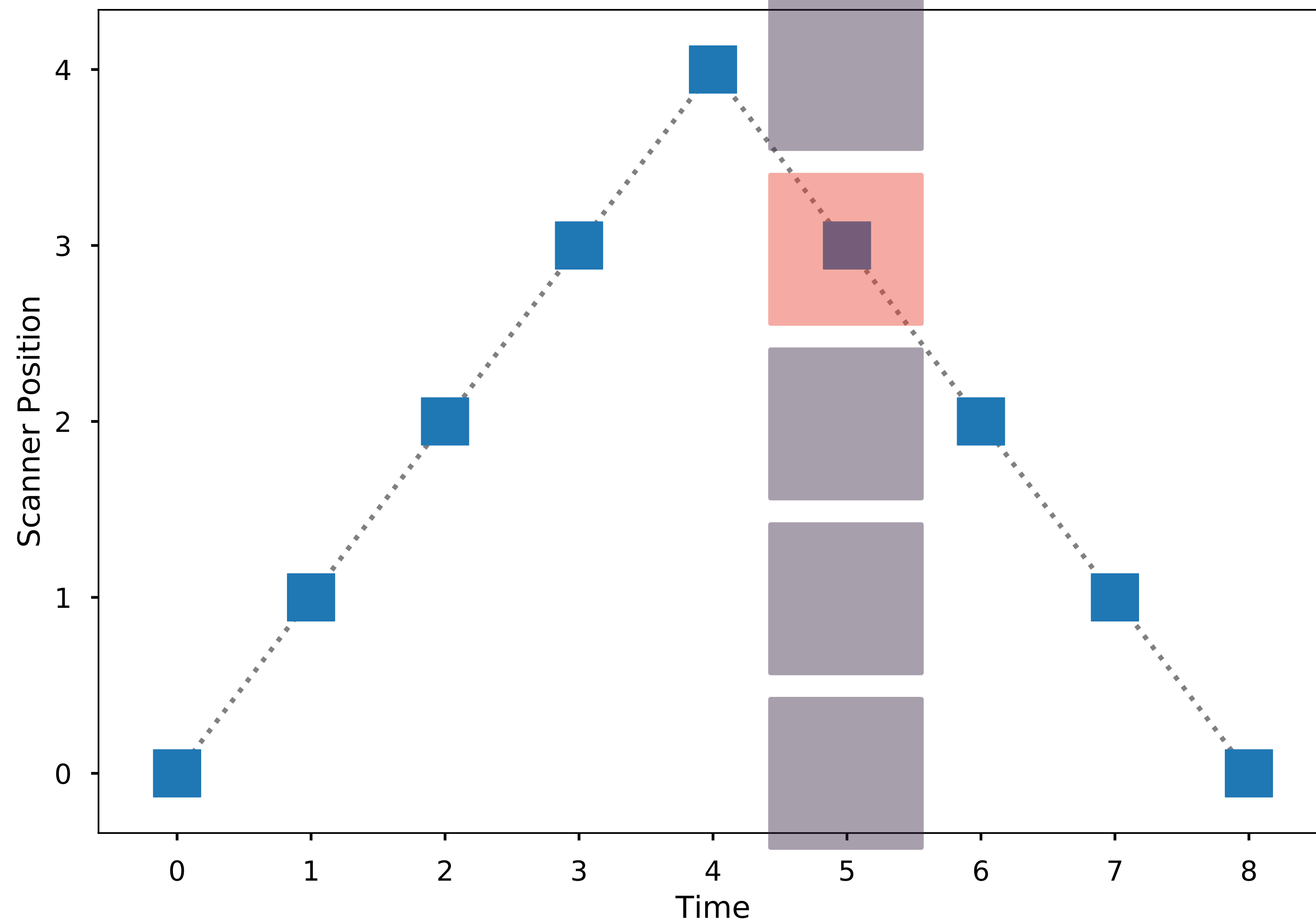
# Math is Magic

---





# Math is Magic



`scanner_height = 5`

`cycle_width = 8`

`time_step = 21`

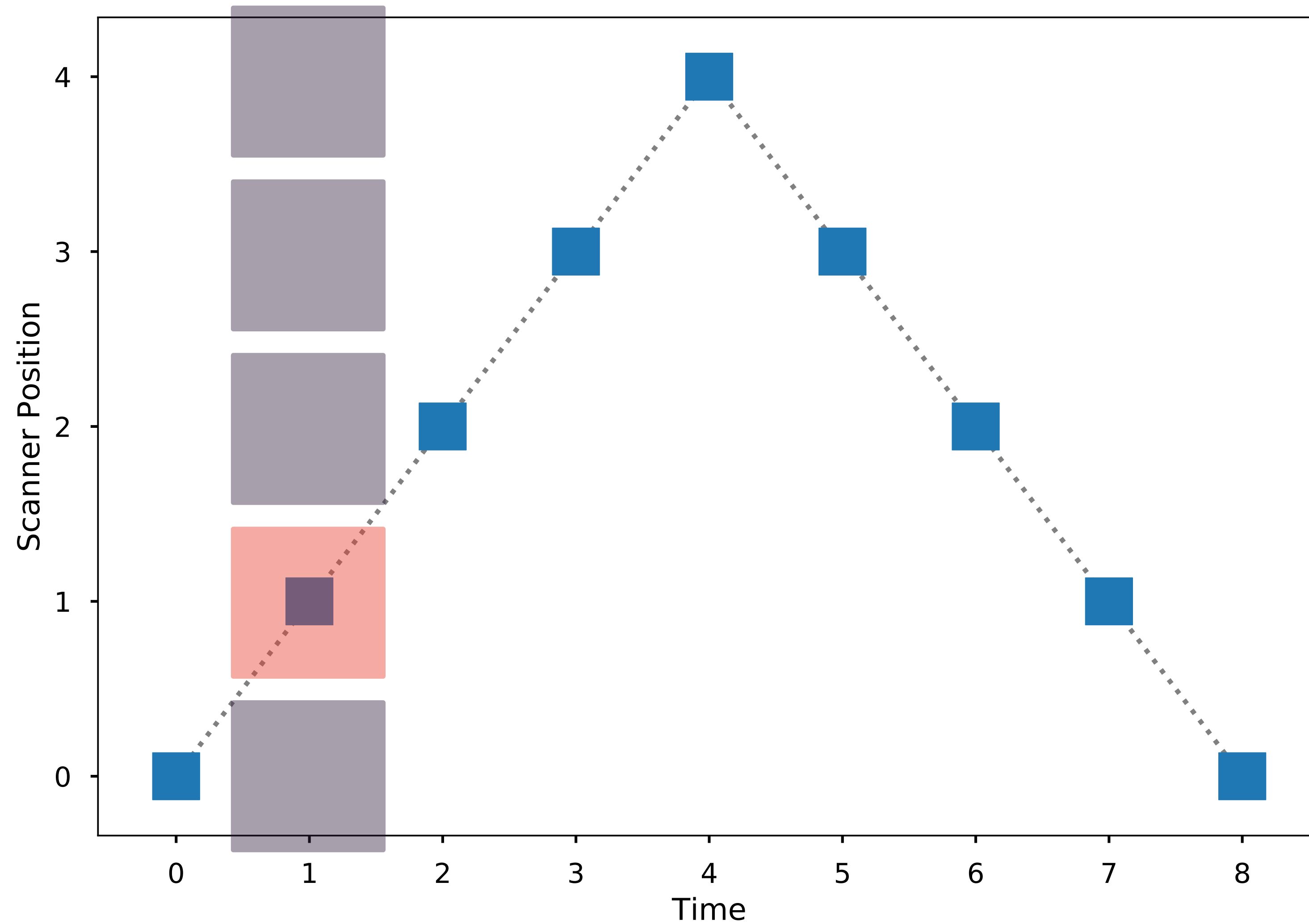
`cycle_pos = time_step % 8`

`→ 5`

`return 8 - 5`

`→ 3`

# Math is Magic



## Second Try (calc\_scanner\_pos)

---

```
def calc_scanner_pos(  
    scanner_height: int, time_step: int) -> int:  
    cycle_midpoint = scanner_height - 1  
    cycle_width = cycle_midpoint * 2  
    cycle_position = time_step % cycle_width  
    return (  
        cycle_position  
        if cycle_position <= cycle_midpoint  
        else cycle_width - cycle_position)
```

## Second Try (firewall)

---

```
firewall = {scanner_slot: scanner_height}
```

## Second Try (caught\_crossing)

---

```
def caught_crossing(firewall: dict, width: int, t_start: int) -> bool:
```

```
    for pos in range(width):
```

```
        if pos in firewall:
```

```
            scanner_height = firewall[pos]
```

```
            scanner_pos = calc_scanner_pos(scanner_range, t_start + pos)
```

```
            if scanner_pos == 0:
```

```
                return True
```

```
    return False
```

**Is there a scanner?**



**Is the scanner at the bottom?**





## Second Try (find\_start)

---

```
def find_start(firewall: dict) -> int:
```

```
    width = max(firewall.keys()) + 1
```

```
    for t_start in itertools.count(0):
```

```
        if not caught_crossing(firewall, width, t_start):
```

```
            break
```

```
    return t_start
```

**Infinite loop**



**Did we get caught?**



**Break out if we didn't get caught**



**Did It Work?**

**Answer: 3,823,370 time-steps**

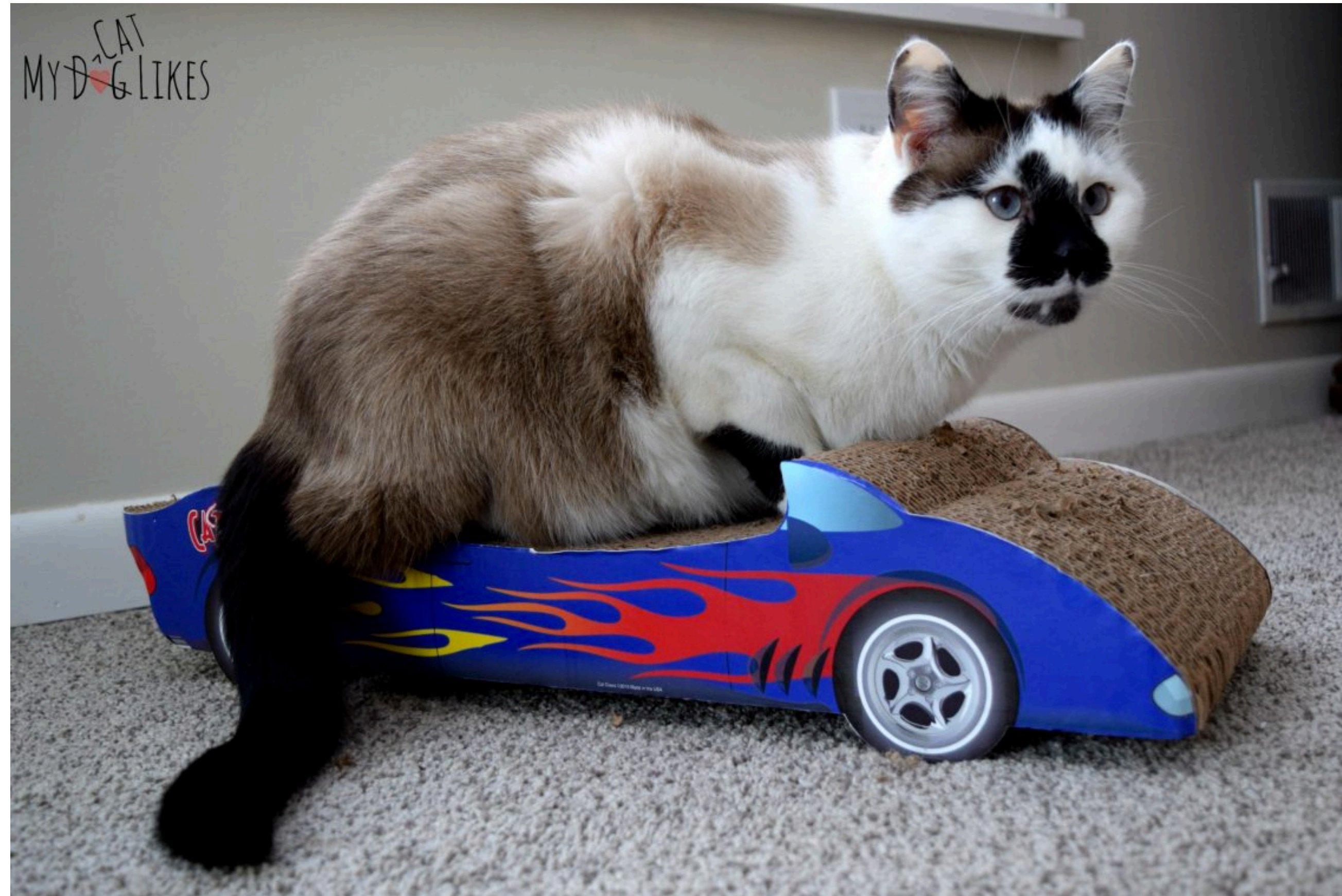
**How Long Did it Take?**

**~6 seconds**



# Cool, cool. Can we go faster?

---



<https://mycatlikes.com/cat-claws-convertible-cat-scratcher-review/>

# Results

	Time (milliseconds)
First Try	600,000
Second Try	6,000
PyPy	600
Numba	2
Cython	1

# Array Firewall

---

**firewall = [3, 2, 0, 0, 4, 0, 4]**



**Numba**

**Compiles Python on-the-fly**

**100% plain Python**

**Used with NumPy for maths**

# Numba @jit

---

**“just in time”**

**input types**

`@jit(boolean(int32[:], int32, int32), nopython=True)`

**return type**

**fast mode**



# Numba (caught\_crossing)

---

## Numba decorator

`@jit(boolean(int32[:], int32, int32), nopython=True)`

`def caught_crossing(`

`firewall: np.array, width: int, t_start: int) -> bool:`

`for pos in range(width):`

`if firewall[pos] != 0:`

`scanner_pos = calc_scanner_pos(firewall[pos], t_start + pos)`

`if scanner_pos == 0:`

`return True`

`return False`

**Is there a scanner?**

**Is the scanner at the bottom?**

# Numba (find\_start)

---

```
@jit(int32(int32[:]), nopython=True)
def find_start(firewall: np.array) -> int:
```

```
    t_start = 0
```

```
    width = len(ranges)
```

```
    while True:
```

```
        if caught_crossing(ranges, width, t_start):
```

```
            t_start += 1
```

```
        else:
```

```
            break
```

```
    return t_start
```

**Infinite loop**



**Did we get caught?**



**Break out if we didn't get caught**



**Cython**

**Extended Python-like language**

**Compiles to C**

**Great for wrapping C libraries**

# Cython (caught\_crossing)

---

```
@cython.boundscheck(False) ← turn off safety checks
@cython.wraparound(False)
cdef bint caught_crossing(int[:] firewall, int width, int t_start):
    cdef int pos, scanner_pos, scanner_range
    for pos in range(width):
        if firewall[pos] != 0:
            scanner_pos = calc_scanner_pos(firewall[pos], t_start + pos)
            if scanner_pos == 0:
                return False
    return True
```

**types on everything**

# Results

	Time (milliseconds)
First Try	600,000
Second Try	6,000
PyPy	600
Numba	2
Cython	1

# THANK YOU!!!

(tweet me: @jiffyclub)

1. **Collect Data**
2. **Analyze**
3. **Experiment**
4. **GOTO 1**

[github.com/jiffyclub/pycon-2018-talk](https://github.com/jiffyclub/pycon-2018-talk)



<http://www.pusheen.com/post/95781992231> <https://imgur.com/t/pusheen/P1Fvz>