

CSCI 141

Scott Wehrwein

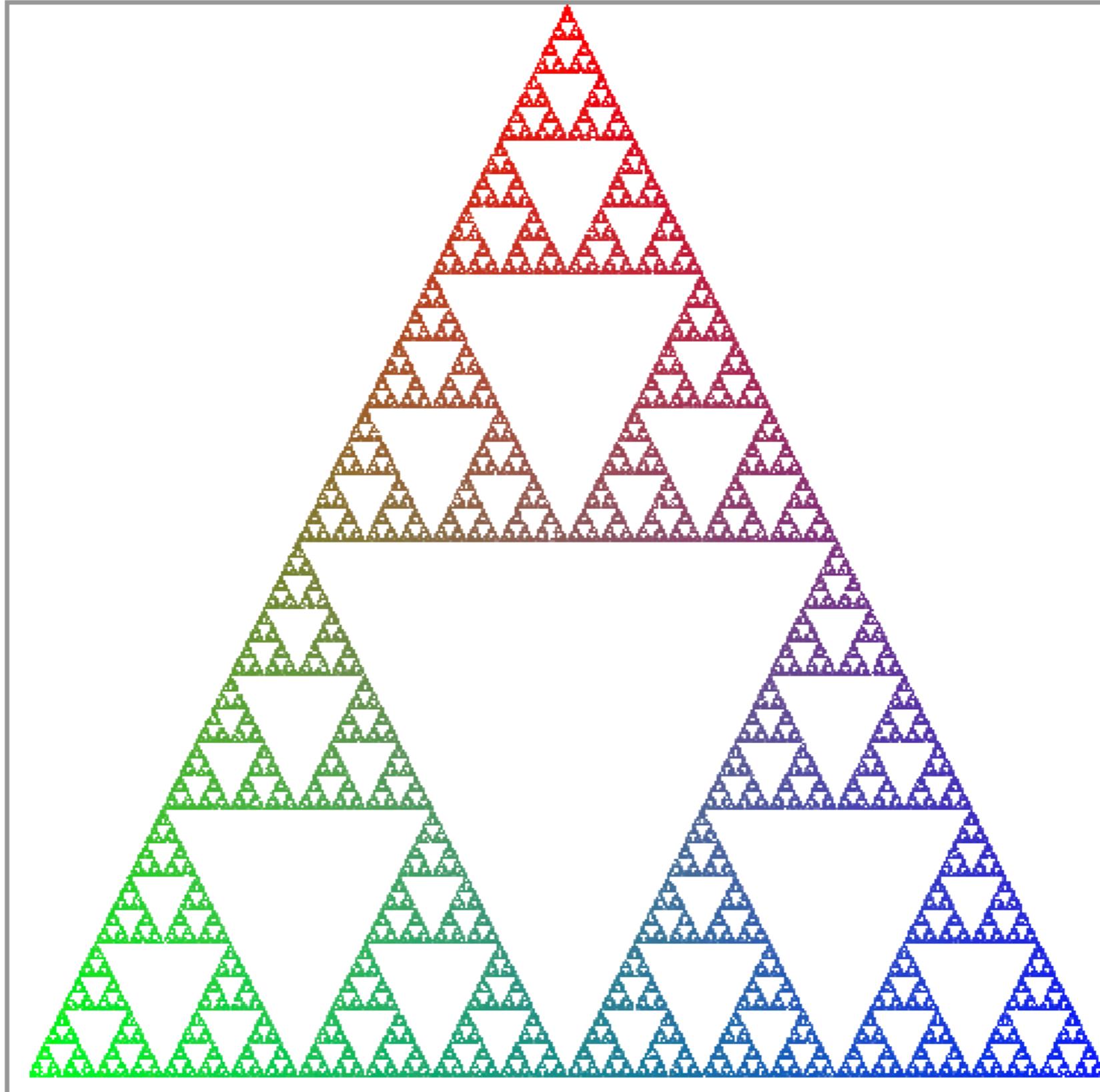
(Tuples,)

Goals

- Understand the basic usage of **tuples**:
 - using tuples to return multiple values from a function
 - **packing** and **unpacking** via the assignment operator

A4

Your task:
Draw this.



A4: Pseudocode

```
# Let p be a random point in the window
# loop 10000 times:
#     c = a random corner of the triangle
#     m = the midpoint between p and c
#     choose a color for m
#     color the pixel at m
#     p=m
```

Strategy: break this down into manageable pieces by **inventing** functions that solve pieces of the problem!

A4: Pseudocode

```
# Let p be a random point in the window
# loop 10000 times:
#     c = a random corner of the triangle
#     m = the midpoint between p and c
#     choose a color for m
#     color the pixel at m
#     p=m
```

Midpoint Function

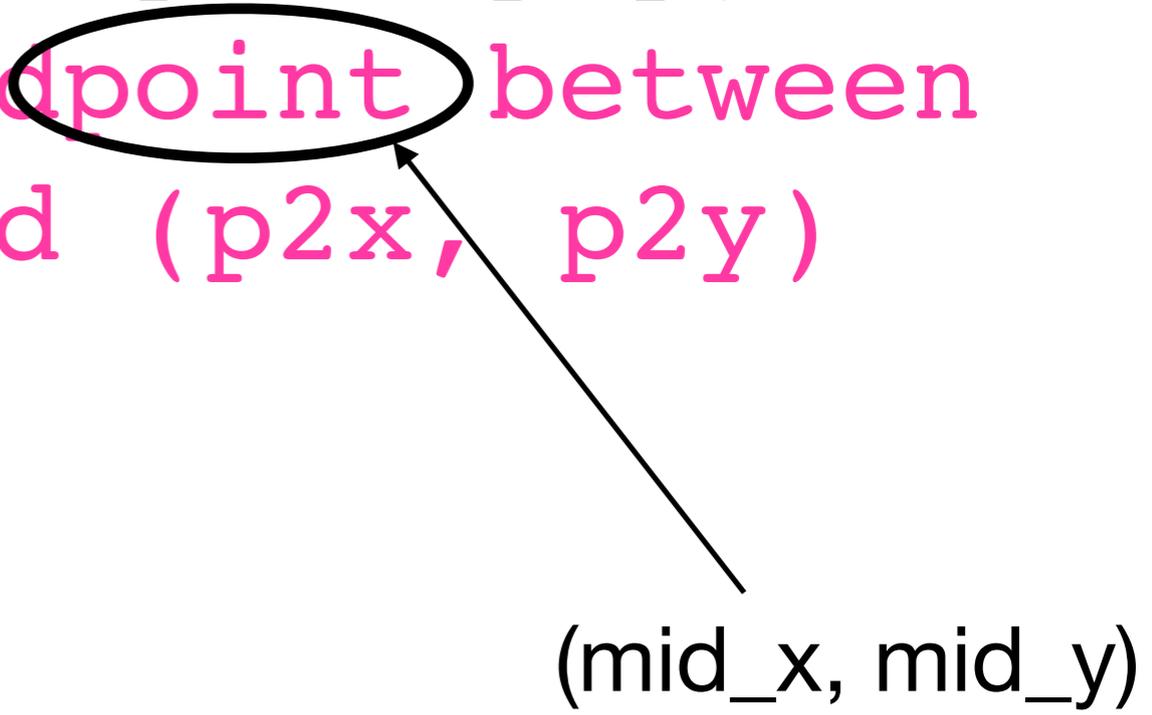
```
def midpoint(p1x, p1y, p2x, p2y):  
    """ Return the midpoint between  
        (p1x, p1y) and (p2x, p2y)  
    """  
    # code here
```

Midpoint Function

```
def midpoint(p1x, p1y, p2x, p2y):  
    """ Return the midpoint between  
        (p1x, p1y) and (p2x, p2y)  
    """  
    # code here
```

Midpoint Function

```
def midpoint(p1x, p1y, p2x, p2y):  
    """ Return the midpoint between  
        (p1x, p1y) and (p2x, p2y)  
    """  
    # code here
```



The word "midpoint" in the docstring is circled in black. An arrow points from the text "(mid_x, mid_y)" below to the circled word "midpoint".

(mid_x, mid_y)

Midpoint Function

```
def midpoint(p1x, p1y, p2x, p2y):  
    """ Return the midpoint between  
        (p1x, p1y) and (p2x, p2y)  
    """  
    # code here
```

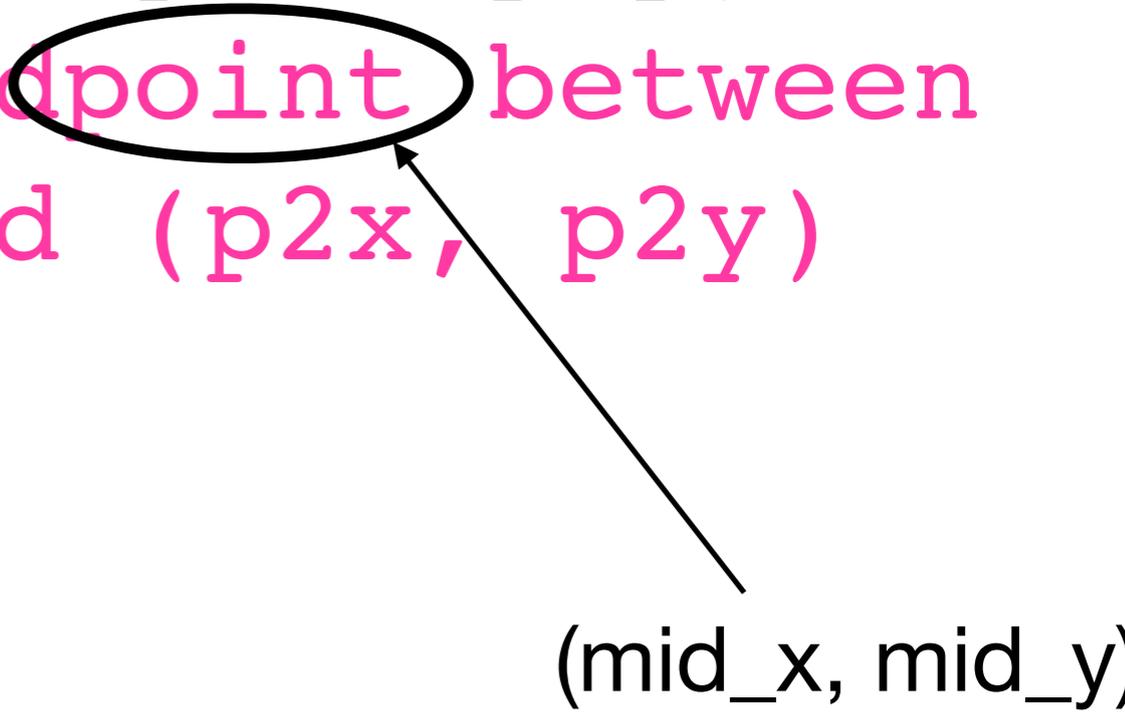
(mid_x, mid_y)

This is **two**
things!?
Can we return
two things?



Midpoint Function

```
def midpoint(p1x, p1y, p2x, p2y):  
    """ Return the midpoint between  
        (p1x, p1y) and (p2x, p2y)  
    """  
    # code here  
    # mid_x = . . .  
    # mid_y = . . .  
  
    return mid_x, mid_y
```



The word "midpoint" in the docstring is circled in black. An arrow points from this circle to the text "(mid_x, mid_y)" on the right side of the slide, indicating that the function returns a tuple of the midpoint coordinates.

Returning Multiple Values

- You can return multiple values from a function by grouping them into a comma-separated sequence:

```
return mid_x, mid_y
```

- You can assign each to a variable when calling the function like this:

```
mx, my = midpoint(p1x, p1y, p2x, p2y)
```

These are actually tuples

- A tuple is a sequence of values, optionally enclosed in parens.


(of any types!)

```
(1, 4, "Mufasa")
```

- You can “pack” and “unpack” them using assignment statements:

```
v = (1, 4, "Mufasa") # packing
```

```
(a, b, c) = v # "unpacking"
```

These are actually tuples

- Tuples can also be passed *into* functions as arguments:

```
def midpoint(p1, p2):  
    """Compute the midpoint between p1 and p2"""  
    p1x, p1y = p1  
    p2x, p2y = p2  
  
    # . . .  
    # return mx, my
```

Tuples: Demo

Tuples: Demo

- assignment, packing, unpacking
- with and without parens (printing)
- swapping
- equality
- mismatched # values to unpack

Tuples - 1

a = 1

b = 2

c = 3

v = (a, a, c)

print(v, sep=" ")

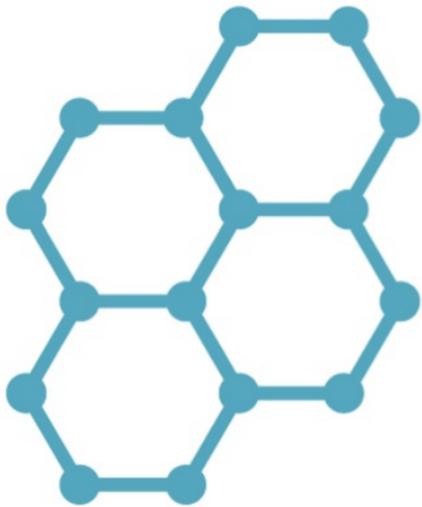
What does this print?

A: 1 2 3

B: 1 1 3

C: (1, 2, 3)

D: (1, 1, 3)

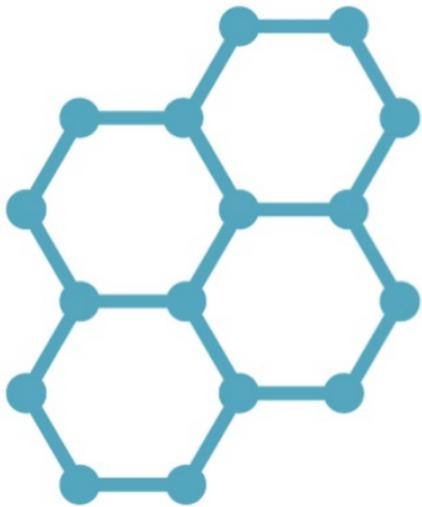


Tuples - 2

```
a = 1
```

```
b = 2
```

```
c = 3
```



```
a, b, c = (a, a, c)
```

```
print(a, b, c, sep=" ")
```

```
# What does this print?
```

```
# A: 1 2 3
```

```
# B: 1 1 3
```

```
# C: (1, 2, 3)
```

```
# D: (1, 1, 3)
```

Midpoint Function

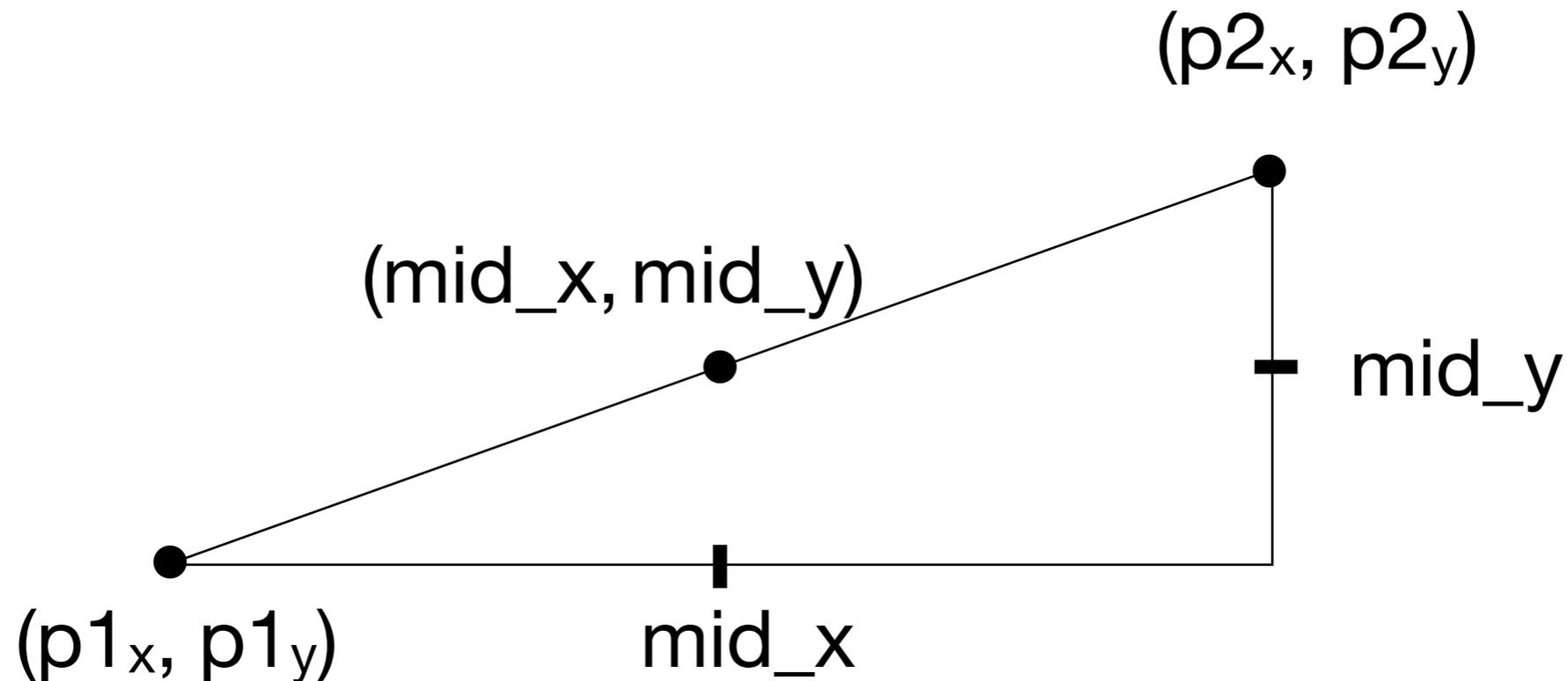
```
def midpoint(p1x, p1y, p2x, p2y):  
    """ Return the midpoint between  
        (p1x, p1y) and (p2x, p2y)  
    """  
  
    # code here  
    # mid_x = . . .  
    # mid_y = . . .  
  
    return mid_x, mid_y
```

Midpoint Function

$mid_x = \dots$

$mid_y = \dots$

Okay, but how do you actually calculate this?



$$mid_x = (p1_x + p2_x) / 2$$

$$mid_y = (p1_y + p2_y) / 2$$

Demo: writing the midpoint function

- With tuple as return value
- Switch to tuples as parameters for points