

CSC 111:

Intro to Computer Science through Programming

Spring 2017
Prof. Sara Mathieson



Notecards

Write down any topics you would like to cover next week during review

Admin

- + No more help on final projects (I'll answer Piazza questions through the end of today)
- + After today, only Piazza questions on class content in preparation for the final exam (asking about past homeworks, labs, etc is fine)
- + **Final project** is due May 3 (Wednesday)
- + TA hours next week: only final review help (Sun-Wed)
- + TA review session: next Wed (May 3)
- + Self-scheduled **final exam** (similar style to the midterm)
- + **Last office hours: Monday 3-5pm (final prep only)**

Outline: 4/28

- + Finish discussion on sort and search
- + Tuples
- + Final project tips
- + Quiz 9
- + Shuffle sort and discussion of efficiency

Finish sort and search

Future work: how to actually get the closest element returned?

- + Example: when I search for "Bel", the return value is "Astou".
- + "Astou" < "Bel" < "Belise", but depending on how we split the list, we will not get "Belise" as the answer
- + Any solution ideas?

Future work: how to actually get the closest element returned?

- + Example: when I search for "Bel", the return value is "Astou".
- + "Astou" < "Bel" < "Belise", but depending on how we split the list, we will not get "Belise" as the answer
- + Any solution ideas?
- + One idea: have an additional base case for lists of length 2, then compare the query with each of these 2 elements to find the closer one

Future work: how to actually get the closest element returned?

- + Example: when I search for "Bel", the return value is "Astou".
- + "Astou" < "Bel" < "Belise", but depending on how we split the list, we will not get "Belise" as the answer
- + Any solution ideas?
- + One idea: have an additional base case for lists of length 2, then compare the query with each of these 2 elements to find the closer one
- + Another idea: compare the query to both middle elements (right before and right after the split). If the query is between them, use the closer one to determine left/right recursion

Number of comparisons for search

- + Say our list is of length n
- + What is the worst case number of comparisons for linear search? (i.e. start at the beginning of the list and go through until we find the closest one)

Number of comparisons for search

- + Say our list is of length n
- + What is the worst case number of comparisons for linear search? (i.e. start at the beginning of the list and go through until we find the closest one)

Have to go to the end of the list: n comparisons $\Rightarrow O(n)$

Number of comparisons for search

- + Say our list is of length n
- + What is the worst case number of comparisons for linear search? (i.e. start at the beginning of the list and go through until we find the closest one)

Have to go to the end of the list: n comparisons $\Rightarrow O(n)$

- + What is the worst case number of comparisons for binary search?

Number of comparisons for search

- + Say our list is of length n
- + What is the worst case number of comparisons for linear search? (i.e. start at the beginning of the list and go through until we find the closest one)

Have to go to the end of the list: n comparisons $\Rightarrow O(n)$

- + What is the worst case number of comparisons for binary search?

$$\frac{n}{2^x} = 1$$

Number of comparisons for search

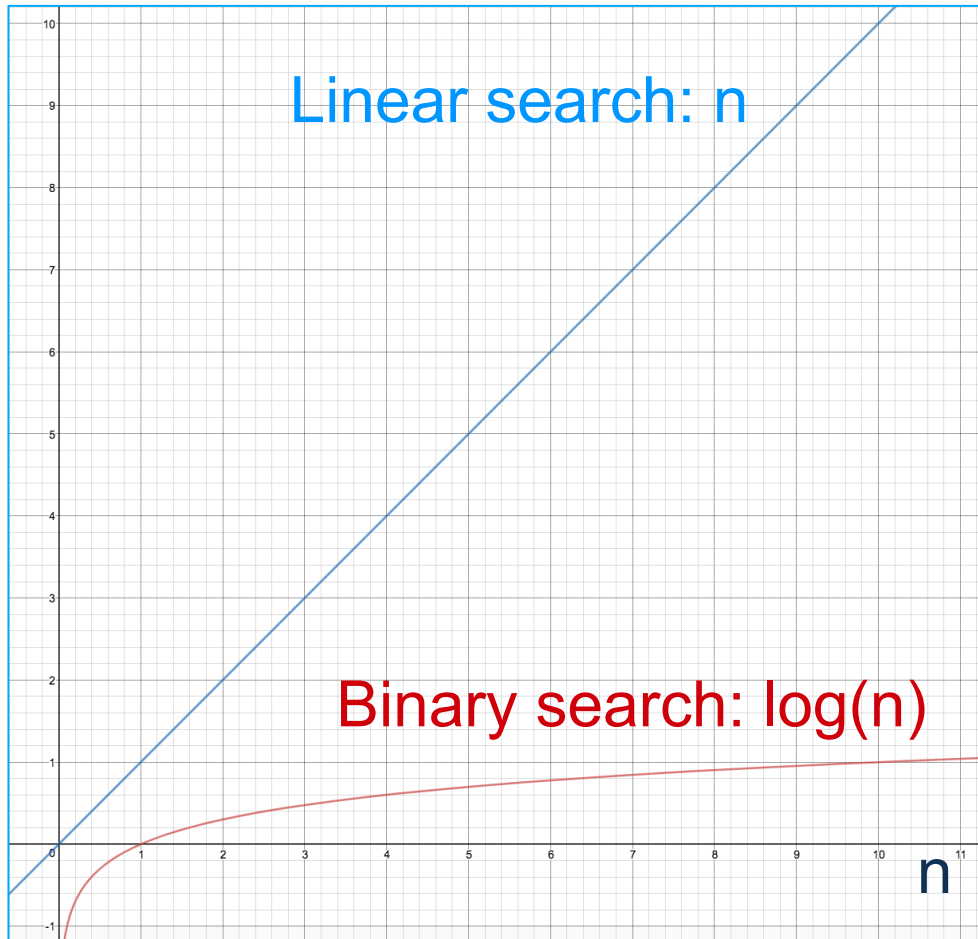
- + Say our list is of length n
- + What is the worst case number of comparisons for linear search? (i.e. start at the beginning of the list and go through until we find the closest one)

Have to go to the end of the list: n comparisons $\Rightarrow O(n)$

- + What is the worst case number of comparisons for binary search?

$$\frac{n}{2^x} = 1 \longrightarrow x = \log_2(n) \longrightarrow O(\log(n))$$

Linear vs. Binary search



Tuples

Tuples: another type

- + **Tuples** are created using `(...)`, kind of like **lists** are created using `[...]`
- + Main difference: tuples are *immutable*
- + `circle = Circle((x,y), radius)` will produce an error
- + Accidentally using a tuple `(x,y)` instead of a `Point(x,y)`
- + Tuples also show up when using simultaneous return
- + Common in other situations as well, whenever commas are used without other info

Common tuple error messages

```
>>> circle = Circle((100,200), 10)
Traceback (most recent call last):
  File "<pyshell#21>", line 1, in <module>
    circle = Circle((100,200), 10)
  File "/Users/ssheehan/Dropbox/Website/smith/spring17/
csc111/lecs/lec37/graphics.py", line 646, in __init__
    p1 = Point(center.x-radius, center.y-radius)
AttributeError: 'tuple' object has no attribute 'x'
>>>
>>>
>>> type((100,200))
<class 'tuple'>
```

Final Project tips

Other terms for instance variable

- + Instance variable
- + Member variable
- + Attribute (see error below)

```
Traceback (most recent call last):  
  File "/Users/ssheehan/Dropbox/Website/smith/spring17/csc111/hws/hw9/hw9.py", line 215, in <module>  
    main()  
  File "/Users/ssheehan/Dropbox/Website/smith/spring17/csc111/hws/hw9/hw9.py", line 208, in main  
    fish.move(width)  
  File "/Users/ssheehan/Dropbox/Website/smith/spring17/csc111/hws/hw9/hw9.py", line 61, in move  
    if self.x < 0:  
AttributeError: 'Fish' object has no attribute 'x'
```

Final project tips

- + Banner class: use `setText(string)`
- + Tracking the image: use `getAnchor()`, which returns a `Point`
- + Use any methods from the `graphics.py` library
- + Now that we've learned many methods and functions, you should be able to use new ones using existing documentation

<http://mcsp.wartburg.edu/zelle/python/graphics/graphics.pdf>

Quiz 9

Question 1

+ **getMouse()** pauses execution: less helpful for animations where you always want some movement

Question 1

Correct

1.00 points out of
1.00

Which of these statements is accurate when it comes to `getMouse` versus `checkMouse` ?

Select one:

- Both `getMouse` and `checkMouse` return the point where the mouse was clicked (if it was clicked) as a `Point()` object. ✓
- Both `getMouse` and `checkMouse` pause execution, waiting for the user to click the mouse in the window.
- Both `getMouse` and `checkMouse` return `None` under certain circumstances.

Your answer is correct.

The correct answer is: Both `getMouse` and `checkMouse` return the point where the mouse was clicked (if it was clicked) as a `Point()` object.

Question 2 (dropped)

Question 2

Correct

1.00 points out of

1.00

```
class MSDie:
    def __init__( self, num_sides ):
        self.sides = sides
        self.value = 1
        random = random.randint(1, self.sides)

    def roll( self ):
        self.value = random

die = MSDie(8)
die.roll()
print(die.value)
```

What's wrong with this piece of code?

Select one:

- missing a return statement
- wrong parameters passed in
- `random` can not be accessed because it's a local variable ✓
- `value` is not an instance variable

Your answer is correct.

The correct answer is: `random` can not be accessed because it's a local variable

Question 3

Question 3

Correct

1.00 points out of
1.00

You spontaneously decide that you despise for-loops. You're now on a mission to refactor any for-loop you encounter to a while loop. Your first enemy is the for-loop below.

```
b = 6
for i in range(10):
    if i % b == 0:
        print('found one')
        break
```

What would the condition for the while loop be if the structure looks like this?

```
while <condition>:
    i++
    print('found one')
```

Should be `i += 1`

Select one:

- `i % b == 0 and i < 10`
- `i > 10`
- `i++ > 10`
- `i <= 10`
- `i % b != 0 and i < 10` ✓

Your answer is correct.

The correct answer is: `i % b != 0 and i < 10`

Shuffle Sort

Demo