

# CS 150: Review

Cynthia Taylor  
Oberlin College  
May 5<sup>th</sup> 2014

# Open Discussion on Community and Academic Environments in the Math and CS Departments

- 12:15pm on Thursday May 8<sup>th</sup>
- Free pizza if you sign up in advance at <http://bit.ly/womacs>.
- A conversation about how to make the Math and CS departments more accessible to all.
- Hosted by Women in Math and Computer Science

# What will this print?

```
def R(s) :  
    if len(s) == 0 :  
        return ""  
    else :  
        return s[0] + R(s[1:]) + s[0]  
  
print(R("cat"))
```

A. taccat  *$s[-1] + R(s[:-1])$*  E. I don't know

B. tacat

C. cattac

D. catac

*- base case 1*

Write a recursive function `lessThan(L,e)` that takes a list `L` and an element `e`, and returns `True` if all of the elements of `L` are less than `e`, and `False` otherwise. You cannot make any assumptions about what type of objects `e` and the elements of `L` are, but you can assume that `<`, `>`, `=` etc work. Do not use loops.

# What is wrong with this code?

```
def __init__(a, b, c):  
    a = self.a  
    b = self.b  
    c = self.c
```

*self*  
*self.a = a*  
*not defined*  
*self.num = a*

Suppose A, B and C are boolean variables. Write a boolean expression that evaluates to true if and only if one or more of these variables are False.

A. (not A) and (not B) and (not C) - all need to be false

B. (not A) or (not B) or (not C)

C. not (A and B and C)

D. More than one of the above

E. I don't know

# What is the Big O run time?

```
def s(L) :
```

```
    x=0
```

```
    for i in range(len(L)) :
```

```
        for j in range(50) :
```

```
            L[i] = L[i] + j
```

```
    for i in range(len(L)) :
```

```
        print(L[i])
```

A.  $O(1)$

B.  $O(n)$

C.  $O(n^2)$

D.  $O(n^3)$

E. I don't know

$O(n^2)$

# Which has the best big O run time?

A. Selection Sort

B. Insertion Sort

C. Bubble Sort

D. They are all the same

E. I don't know

for  $i$  in  $\text{Len}(L)$ :

while  $L[i]$  unsorted

move stuff  
a count

merge sort -  $O(n \log n)$



# Next Time

- Review
- Final Exam:
  - Wednesday the 14<sup>th</sup>, 7pm