CS5 Today: So. Much. Recursion.

I feel rec’d...
Reminder of your Challenge:
Learn names!

Stand up – and introduce yourself to THREE people to your left!

Around the room: “This is X and Y and Z and I am A.”
```python
def keepvwl(S):
    if len(S) == 0:
        return ''
    elif S[0] in 'aeiou':
        return S[0] + keepvwl(S[1:])
    else:
        return '' + keepvwl(S[1:])
```

Last time:

`abcd`
# recursive tower

def tower(n):
    if n == 0:
        return 1
    else:
        return 2 * tower(n-1)

tower(0) = ?

tower(3) = 2^2^2

tower(4) = 2^2^2^2

tower(5) = 2^2^2^2^2

inductive definition:

inductive case:
    tower(n) = 2 \cdot tower(n-1)

base case:
    tower(0) = ?
Deep-Reversing a List

>>> reverse([1, [2, [4, 5], 6], 7])
[7, [2, [4, 5], 6], 1]

>>> deepReverse([1, [2, [4, 5], 6], 7])
[7, [6, [5, 4], 2], 1]
Merging Two Lists

In [1]: merge([5, 10], [2, 12])
Out[1]: [2, 5, 10, 12]

In [2]: merge([2, 7], [2, 6])
Out[2]: [2, 2, 6, 7]

In [3]: merge([2, 7], [])
Out[3]: [2, 7]

In [4]: merge([], [42, 50])
Out[4]: [42, 50]
Mapping a function to a list

```python
def dbl(x):
    return x * 2

In [1] map(dbl, [1, 2, 3, 4])
Out [1][2, 4, 6, 8]
```
Reducing a list

def add(x, y):
    return x + y

In [1] reduce(add, [1, 2, 3, 4])
Out[1] 10