15-110 recitation 07

Recap

- recursion on lists
- binary search
- big-O

Reminders!

HW 3 due Monday!

Reach out to your TA's for small groups and 1v1!

Problems

code trace

Problem

```
def mystery(a, b):
    # assume b is non-negative integer
    if (b == 0):
        return 1
    else:
        return a * mystery(a, b-1)

print(mystery(2,5))
```

What familiar function does mystery compute?

listSum

Write a function that takes in a list and returns the sum of the elements in the list.

linear search v. binary search

Problem

Run through the simulation here

```
def pow(m, n) {
    ret = 1
    for i in range(n):
        ret = ret * m
    }
    return ret
}
```

Run the code trace on the list [2, 4, 6, 7, 10, 11] and find 1, 12, and 6.

What is the complexity of pow in terms of m and n? How many steps does this function take?

How many steps are required to search for 11 in linear search? Binary search?

During binary search, how many elements (maximum) are checked in a list of length *n*?