The Beauty & Joy of Computing
Lecture #8
Recursion

GO SEE INCEPTION!
The coolest movie last year highlights recursion, and it was up for best picture. If you haven’t seen it yet, you should, because it will help you understand recursion!!

New Rule: Use scratch paper in lab!
The problems there are hard enough that you won’t be able to keep it in your head.

You already know it!

Trust the Recursion

When authoring recursive code:
- The base is usually easy: “when to stop?”
- In the recursive step
  - How can we break the problem down into two:
    - A piece I can handle right now
    - The answer from a smaller piece of the problem
  - Assume your self-call does the right thing
  - How to combine parts to get the overall answer?
- Practice will make it easier to see idea

Sanity Check…

Recursion is ≠ Iteration (i.e., loops)
- Almost always, writing a recursive solution is ≠ than an iterative one
  a) more powerful than, easier
  b) just as powerful as, easier
  c) more powerful than, harder
  d) just as powerful as, harder

Summary

Behind Abstraction, Recursion is probably the 2nd biggest idea about programming in this course
- It’s tremendously useful when the problem is self-similar
- It’s no more powerful than iteration, but often leads to more concise & better code