Introduction to: Computers & Programming: Recursion

Adam Meyers
New York University
Summary

• What is recursion?
• Some Simple Examples
• Some Rules about Recursion
• The equivalence between recursion and while loops
• Complex problems/puzzles that are easier to understand through recursion
What is Recursion?

• A recursive function is a function that calls itself to solve successively smaller versions of the same problem.

• Recursive functions typically divide problems into:
  – One or more base cases which have simple solutions.
  – All other cases, for which the function must call itself on smaller instances of the problem
  – Without a base case, recursion is endless (like while loops)

• Recursive functions reduce all complex problems to instances of the base case + additional stuff + a recursive step

• Any instance of recursion can be converted to iteration (loop-based solution).

• Some problems are more naturally implemented recursively and others are more naturally implemented as iteration.
Recursion vs. Iteration

• Examples from recursion-functions.py
  – Counting
  – Counting down
  – Factorial
  – The fibonacci Sequence
  – Endless timer

• For each of these, which solution is more intuitive?
The Strategy for Solving a Problem with Recursion

• Given a big problem, find a problem that is exactly one step smaller.

• Assume that you can handle the smaller problem and figure out what more you would need to do to solve the whole problem.

• Identify the smallest problem of this type that has a very simple answer and does not require any further breaking down – this is the base case.

• If all problems of this type larger than the base case can be broken down this way, recursion should work.
Class Problem 1 (10 minutes)

• Write a recursive function for computing compound interest.
• The function should take three arguments:
  – The principle
  – Interest per interval
  – The number of intervals
• It should return the total after adding all the interest
• For example, 10% on 1000 compounded daily
  – Principle = 1000
  – Interest = .1 / 360
  – Intervals = 360
• The total should be about: 1105.16
  – Don't use the shortcut \((1 + 1/\text{interest})^{\text{intervals}}\)
Class Problem 2 (15 minutes)

• Given a list of random numbers, find the highest number
• Base case 1: If there are two numbers in the list, use >= to find the highest number
• Base case 2: If there is only one number in the list, that number is the highest number
• For all other cases:
  – Divide the list into 2 parts that are equal in size or one away from being equal
  – Find the highest value from each list:
  – Compare them (base case 2)