Lab08 Recursion (10.21.19) CS103 Fall 2019

author john k johnstone jkj at uab dot edu
course CS103 Fall 2019
license MIT
version Fall 2019

materials

- *lab08_ 19fa103.pdf* (this document)
- *lab08_ 19fa103.py* (with functions you will implement)
  I don't expect you to finish all these functions today: good exercises for outside lab too
- *lab08_ tests_ 19fa103.py* (create your own test calls here, see lab07)
- lab partner: same lab partner as last week

purpose

- learn recursion

A recursive function has two main parts:
at least one base case, tested before the recursive call, and at least one recursive call.

base case the smallest, trivial instance of the input (e.g., 1+. . . +n for n=1)
recursive call of the function f a call of f inside the body of f

The general structure of a recursive function is as follows:

```
- (if necessary) assertion that input is correct, to avoid an infinite loop
- if the input is a base case, solve it directly
- if the input is not a base case, recursively call on a smaller input,
  and merge the rest of the input into the result of this recursive call
```

The recursive factorial code discussed on Thursday follows this model exactly:

```
def factorialRecur (n):

    assert n >= 0
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorialRecur (n-1)
```

Note that we are expecting a gentle version of recursion here, as introduced in lecture and described above, with recursive call of the original function. There is a slightly more advanced (and robust) version of recursion called tail recursion, which we will discuss later.

in-class exercise

1+2+. . . +n recursively

exercises

- string reversal, recursively
- string reversal, recursively, using another approach
- sum from M to N, recursively

(congratulations if you get this far!)

- given a list L, is L a list of integers, solved recursively?
- argmax, recursively

challenges

- generate permutations, recursively
- fix the problem with our recursive implementation of Fibonacci

deliverables

C attendance
B attendance and in-class exercise (1-parameter sum)
A B exercises + string reversal both ways + 2-parameter sum
A+ no A+ option this week (but do have fun with the challenges, just not graded)