CS8 Lab03

Shiliang Tang

04/21/2015
Basic Information

• Prof. Costanzo’s Website

• My personal CS8 homepage:
    • Download lab slides here
    • Lab02 sample code here
    • Project1 sample code here
Lab03

• Lab03 requirements
  – Pair-programming
  – Step 3: write a factorial function, understand accumulator pattern
    • mandatory
    • write code is easy, please make sure you understand
  – Step 4: write a function to calculate terms of Fibonacci sequence
    • must try, may not finish
    • use accumulator pattern
Accumulator Pattern

• Accumulator pattern
  – is a programming pattern in which a final answer is built a piece at a time in a loop.
  – Initialize accumulator(s)
    • \( \text{result} = 1 \)
  – Loop through steps
    • for \( i \) in \( \text{range}(2,n+1) \):
  – Keep update accumulator(s)
    • \( \text{result} = \text{result} \times i \)
• **Lab03 requirements**
  
  – Step 4: some mathematical analysis

<table>
<thead>
<tr>
<th></th>
<th>factorial</th>
<th>fib</th>
</tr>
</thead>
<tbody>
<tr>
<td>initialize</td>
<td>$f_1 = 1$</td>
<td>$f_1 = 1, f_2 = 1$</td>
</tr>
<tr>
<td>iteration</td>
<td>$f_i = i \times f_{i-1}$ (n≥2)</td>
<td>$f_i = f_{i-1} + f_{i-2}$ (n≥3)</td>
</tr>
</tbody>
</table>

• For factorial, in each step of the for loop, the value of the accumulator equals to $f_i$

• Similar, for fib, in each step of the for loop, the values of the 2 accumulators equals to $f_i$ and $f_{i-1}$, respectively
Lab03

• Requirements
  – Step 4: What we don’t expect, although correct

```python
# fib - returns nth term of Fibonacci sequence:
#       1, 1, 2, 3, 5, 8, ... So fib(6) = 8
def fib(n):
    if n == 1 or n == 2:
        return 1
    else:
        return fib(n-1) + fib(n-2)

• Called recursive pattern (not accumulative pattern)
```
Lab03

• Lab03 requirements
  – Step 5: optional
    • hint: try while loop
      – 1. check if the boolean expression is true
      – 2. execute iteration body
      – until the boolean expression is false

– Get your work checked off
  • After you finish step 3 and tried on step 4
  • Do not be too late
  • Submit your Hw3, get your Hw2
Extra Challenge

• Approximate and draw “golden spiral”
  – Use turtle module
  – Useful functions:
    • From: docs.python.org/3.4/library/turtle.html
    • `turtle_name.pensize(width)`
      – Set the line width, set to 5 gives you a thick line
      – If `width` is omitted, the it returns current line width
    • `turtle_name.circle(radius,extent,steps)`
      – Draw a circle, whose radius is `radius`
      – Set `extent` to 180 to draw semicircle, 90 to draw quarter circle
      – `steps` can be omitted