Recursion Lessons – Fall 2021

**Day 1**

* Warmup with students writing (iterative) fact by hand.
* Show answer in Java, run a few times.
* Show relationship between different calls to fact (on slides).
* Introduce idea of recursion (on slides)
* Introduce base case/recursive case.
  + Go through BC/RC for factorial.
  + BC: if n=1, fact(n) = 1.  
    RC: if n>=2, fact(n) = fact(n-1)\*n USE LOCAL VARIABLE.
* Write Java version.
* Do memory diagram for factrec(3).
* Demo turtle graphics.
* Do reverse a string

Day 2

* Go over recursive factorial solution.
* Start with memory diagram for factrec(3), run with debugger at same time.
* Write reverse String

/\*\*  
 \* Reverse a string, recursive version.  
 \* Recursive formulation:  
 \* BASE CASE: If str only has one letter, then the answer is str  
 \* RECURSIVE CASE: If str has more than one letter:  
 \* char lastChar = last character of str  
 \* String smaller = everything in str except the last char  
 \* String revSmaller = reverse(smaller)  
 \* Answer is lastChar + revSmaller  
 \*/

* Then show memory diagram for reverseString("abc")
* Show Fibonacci.??? maybe ???
* Do 4 versions of weird:
* **void** weird(**int** n)  
  {  
   **if** (n == 0)  
   **return**;  
   **else** {  
   weird(n - 1); // optional  
   cout << n << endl;  
   weird(n - 1); // optional  
   }  
  }
  + print, then recurse
  + recurse, then print
  + print, recurse, print
  + recurse, print, recurse
* If time – sum of elements in a list. 🡨 didn't get to. Barely got through part of the 4 print/recurse combos for weird.

Day 3 – Lab Day.

* Did lab 7. Involves drawing recursion diagrams for fact, fib, weird. Then they write 2 recursive functions (power of 2, and sum of ints 1 through n). Then [part c] they do tower of Hanoi.

Day 4 – introduce recursion using arrays and maybe get to binary search.

* Has slides