CS10 : The Beauty & Joy of Computing

Lecture #9 : Recursion

2012-07-03

UC Berkeley EECS
Summer Instructor
Ben Chun

LSO PLAYS IAMUS

Program written at University of Malaga (Study Group on Biomimicry) generated the entire piece – CD coming out in September.

Overview

- Recursion
  - Demo
    - Vee example & analysis
    - Downup
  - You already know it
  - Definition
  - Trust the Recursion!
  - Conclusion

M. C. Escher: *Drawing Hands*

[Image: www.worldofescher.com/gallery/A13.html]
"I understood Vee & Downup"

a) Strongly disagree
b) Disagree
c) Neutral
d) Agree
e) Strongly agree
Definition

- Recursion: (noun) See recursion. 😊

- An algorithmic technique where a function, in order to accomplish a task, calls itself with some part of the task.

- Recursive solutions involve two major parts:
  - Base case(s), the problem is simple enough to be solved directly
  - Recursive case(s). A recursive case has three components:
    - Divide the problem into one or more simpler or smaller parts
    - Invoke the function (recursively) on each part, and
    - Combine the solutions of the parts into a solution for the problem.

- Depending on the problem, any of these may be trivial or complex.
You already know it!

1. There is a little green house.
And inside the little green house
There is a little green house.
And inside the little green house
There is a little green house.

2. \[ n! = n \times (n-1)! \]

3. 

4. A King is a son of a King

5. If all were one

6. What a great man that would be!
And if all men were one man
What a great man that would be!
And if all men were one man
What a great man that would be!

7. What a great man he would be!
And if the great man took the great one
And cut down the great ones
And he left men to grow on
What a great wind that would be!

8. Tree

9. Picture of a hand

10. Picture of a king

11. Family tree of rabbits

12. Tree

13. Tree

14. Mother Goose Shyama
Myself
As I walked by myself
And talked to myself,
M'self said unto me:
"Look to thyself,
For nobody cares for thee,"
I answered myself.
And said to myself:
In the country reported:
"Look to thyself,
Or not look to thyself,
The selfsame thing will be."
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 on a smaller piece of the problem
    - 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

http://xkcd.com/244/

YOUR PARTY ENTERS THE TAVERN.

I GATHER EVERYONE AROUND A TABLE. I HAVE THE ELVES START WHITTLING DICE AND GET OUT SOME PARCHMENT FOR CHARACTER SHEETS.

HEY, NO RECURSING.
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.

http://www.dominiek.eu/blog/?m=200711