Intro to Data Structures

Lecture #12 – Recursion
September 28, 2014

Mark Stehlik

Outline for today

- HW3 demo
- How to test for equality of data fields using the equals() method
- Recursion

Recursion

- Definition
  - See next slide

Recursion

- Definition
  - See previous slide

Recursion

- Definition (for real, this time):
  - A definition that uses the term it is defining in the definition
  - Recursion (n.) - see recursion…
- In computer science:
  - Expressing something in terms of itself (where did we see this done)?
  - A method that calls itself (but, as we saw from the previous slides and the definition above, this can be fraught with peril if not done correctly…)

Linked Lists (recursive definition)

- A linked list is
  - null (the only testable reference value)
  - a node whose next field stores (points to/refers to) a linked list

Method Recursion

- A simple example

```
public static void mystery()
{
    System.out.println("hello");
    mystery();
}
```

- What if I switched the 2 statements?

Recursion (done right)

- So what's all the fuss about recursion?
- A proper recursive method has
  - at least one *base case* (so it stops)
    - the "simplest" version of the problem; solved directly, without recursion
  - at least one *recursive* case (since it is recursive)
    - a recursive call
  - The recursive case(s) need to lead to a base case!
    - a smaller/simpler version of the original problem

Recursion (simple examples)

- Let's write some simple recursive examples…
  - public static void hello(int n)
  - public static int factorial(int n)
  - public static int fibonacci(int n) (but….)

Recursion (recap)

- Start with an *if* (since you have to discriminate between base and recursive cases); no *while*!
- Solve the base case(s) first - these are the simplest cases to solve and are solved directly
- Write recursive calls to deal with smaller versions of general case (sent via parameter(s))
- Statements *before* a recursive call are evaluated on the way down; *after* are evaluated on the way back…

Recursion (etude)

- Iteration (for/while) and recursion are equally expressive (anything you can solve iteratively, you can solve recursively and vice versa)
- BUT, some problems are easier/more efficient to solve iteratively and some, especially those involving recursive data structures, are much easier to solve recursively