Recursion *(cont.)*

CIS 110 Summer 2017 (July 20 Recitation)

Recursion vs. Iteration (loops)

- Iteration: keep repeating until task is complete.
- Recursion: solve a larger problem by breaking it down into smaller pieces until you can solve it and then combine the results.
- A recursive solution is easier to understand and to implement correctly than an iterative solution
- Occasionally (but not always), a recursive solutions is much slower than its iterative counterpart.

Think Recursively

Solving a problem recursively requires a different mindset from solving it iteratively. Some ‘tips’:

1. Pretend to be a little lazy and think about how you could ask others to do your work for you.
2. Find the solution to the simplest/most trivial inputs (the base case(s)).
3. Implement the solution by building on the base cases.

Recursion Exercise 1

public static int largestElement(int[] array)

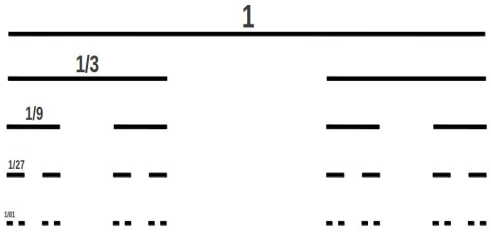
- Return the largest element in “array”
- Hint 1: What is the most trivial type of input? Return the int value Integer.MIN_VALUE for such input. How do we go from a non-trivial “array” to this trivial type?
- Hint 2: Use the max method in Java’s standard Math library, which takes two integers as parameters and returns the larger of the two.

```
int maxValue = Math.max(10, 15); // maxValue is 15
```

Recursion Exercise 2: PennDraw

public static void cantor(int numLevel, double x, double y, double length)

- Recursively draw a representation of the Cantor set as shown below:



- The Cantor set is the set of all real numbers in [0, 1] from which the middle third is deleted in each iteration.
- numLevel: the number of levels of recursion
- x, y: the coordinates of the left endpoint of the line drawn at this level
- length: the length of the line to be drawn at this level