Fundamental Tasks and Techniques - Recursion

Recursion

Binary tree operations which involve both children – e.g. `print`, `size`, `height` – are examples of *recursive* methods.

- the method body involves calls to itself

```
public void print ( TreeNode node ) {
  if ( node.getLeft() == null ) {       // a leaf
    System.out.println(node.getElement());
  } else {                              // internal node
    System.out.println(node.getElement());
    print(tree,node.getLeft());
    print(tree,node.getRight());
  }
}
```

Recursion

Recursion works because of three key features:

- recursive call involves **a smaller version** of the problem
  - e.g. the left and right subtrees are at least one node smaller (and have a height at least one less) than the tree we got
- there is at least one **base case** which is simple enough to solve outright (no recursive calls)
  - e.g. a subtree with just one node (a leaf)
- the base case is **reachable** (so the recursion ends eventually)
  - e.g. the subtrees are smaller (so we progress towards the base case of a leaf) and the children are (only) one level lower than the parent so we can't skip from a taller tree to an empty tree

Recursive Definitions

Many recursive methods come from recursive definitions.

- recursive data structures
  - linked lists
  - binary trees

```
public class ListNode {
  private int element_;
  private ListNode next_;
}
```

- height of a tree

```
height(node) = max(height(node.left),
                   height(node.right))+1
height(leaf) = 1
```

- size of a tree

```
size(node) = size(node.left)+size(node.right)+1
size(leaf) = 1
```

Examples – Recursive Definitions

- factorial

  n! = n (n-1)!
  1! = 1

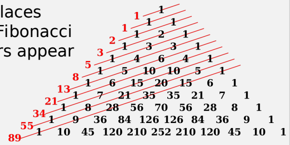
- fibonacci

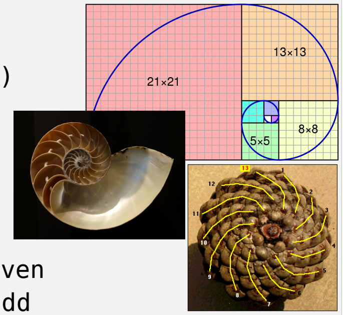
  fib(n) = fib(n-1)+fib(n-2)
  fib(2) = 1
  fib(1) = 1

- exponentiation

  $a^0 = 1$
  $a^n = (a^{n/2})(a^{n/2})$ if n is even
  $a^n = a(a^{n-1})$ if n is odd

some places where Fibonacci numbers appear





https://en.wikipedia.org/wiki/Fibonacci_number
http://www.maths.surrey.ac.uk/hosted-sites/R.Knott/Fibonacci/fibnat.html
https://commons.wikimedia.org/wiki/File:Nautilus_pompilius_-_Fernbank_Museum_of_Natural_History_-_DSC00294.JPG

Examples – Recursive Definitions

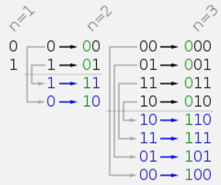
- Euclid's gcd

  gcd = greatest common divisor

  gcd(a,b) = a   if b = 0
  gcd(a,b) = gcd(b,a mod b)   otherwise

- Gray code – successive values differ by only one bit

  1-bit gray code: 0, 1
  n-bit gray code:
    generate (n-1)-bit gray code
    prepend 0 to each element in order
    prepend 1 to each element in reverse order



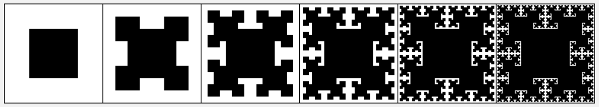
n=1: 0, 1

n=2: 0 → 00, 1 → 01, 1 → 11, 0 → 10

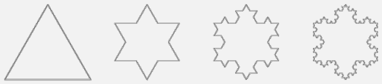
n=3: 00 → 000, 01 → 001, 11 → 011, 10 → 010, 10 → 110, 11 → 111, 01 → 101, 00 → 100

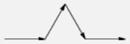
https://commons.wikimedia.org/wiki/File:Binary-reflected_Gray_code_construction.svg

More Recursive Definitions – Fractals



- draw a square half the size of the current region, centered in the region
- repeat with the four quadrants of the current region



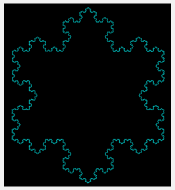
- draw an equilateral triangle
- repeatedly replace each line segment with 

L-Systems

a rule-rewriting system developed by botanist Aristid Lindenmeyer in 1968 to model plant growth processes

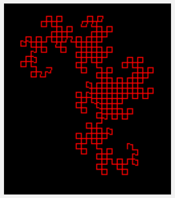
Koch snowflake

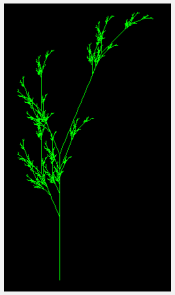
F--F--F--
F → F+F--F+F
angle: 60



dragon curve

F_1
$F_1 \rightarrow F_1+F_2+$
$F_2 \rightarrow -F_1-F_2$
angle: 90

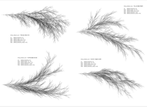




plant

X
X → F-[[X]+X]+F[+FX]-X
F → FF
angle: 22.5

F – move one unit forward, drawing a line
X – do nothing
+, - – turn right, left
[,] – save, restore state



http://blog.kenperlin.com/?p=1453
http://www.erase.net/projects/l-systems/images/page.12.1024.jpg