2.3 Recursion

Overview

What is recursion? When one function calls itself directly or indirectly.

Why learn recursion?
- New mode of thinking.
- Powerful programming paradigm.

Many computations are naturally self-referential.
- Mergesort, FFT, gcd, depth-first search.
- Linked data structures.
- A folder contains files and other folders.

Closely related to mathematical induction.

Greatest Common Divisor

Gcd. Find largest integer that evenly divides into p and q.

Ex: gcd(4032, 1272) = 24.

4032 = 2^5 x 3^2 x 7
1272 = 2^2 x 3^3
gcd = 2^2 x 3 = 24

Applications
- Simplify fractions: 1272/4032 = 53/168.
- RSA cryptosystem.

Euclid's algorithm. [Euclid 300 BCE]

gcd(p, q) = gcd(q, p % q) otherwise

Java implementation.

```java
public static int gcd(int p, int q) {
    if (q == 0) return p;
    else return gcd(q, p % q);
}
```

Greatest Common Divisor

Gcd. Find largest integer d that evenly divides into p and q.

Java implementation.

```java
public static int gcd(int p, int q) {
    if (q == 0) return p;
    else return gcd(q, p % q);
}
```
Towers of Hanoi

Move all the discs from the leftmost peg to the rightmost one.
- Only one disc may be moved at a time.
- A disc can be placed either on an empty peg or on top of a larger disc.

### Towers of Hanoi: Recursive Solution

```java
public class TowersOfHanoi {
    public static void moves(int n, boolean left) {
        if (n == 0) return;
        moves(n-1, !left);
        if (left) System.out.println(n + " left");
        else System.out.println(n + " right");
        moves(n-1, !left);
    }

    public static void main(String[] args) {
        moves(Integer.parseInt(args[0]), true);
    }
}
```

Every other move is smallest disc.
Towers of Hanoi: Recursion Tree

3, true
2, false
1, true
1, true
2, false
1, true
1, true
1, true

1 left
2 right
1 left
3 left
2 right
1 left
1 left

Towers of Hanoi: Properties of Solution

Remarkable properties of recursive solution:
- Takes $2^n - 1$ moves to solve $n$ disc problem.
- Sequence of discs is same as subdivisions of ruler.
- Every other move involves smallest disc.

Recursive algorithm yields non-recursive solution!
- Alternate between two moves:
  - move smallest disc to right if $n$ is even
  - make only legal move not involving smallest disc

Recursive algorithm may reveal fate of world.
- Takes 585 billion years for $n = 64$ (at rate of 1 disc per second).
- Reassuring fact: any solution takes at least this long!

Recursive Graphics

Htree

H-tree of order $n$.
- Draw an H.
- Recursively draw 4 H-trees of order $n-1$, one connected to each tip.

Htree in Java

```java
public class Htree {
    public static void draw(int n, double sz, double x, double y) {
        if (n == 0) return;
        double x0 = x - sz, x1 = x + sz;
        double y0 = y - sz, y1 = y + sz;
        StdDraw.line(x0, y, x, y);
        StdDraw.line(x0, y0, x0, y1);
        StdDraw.line(x1, y0, x1, y1);
        StdDraw.line(x1, y, x1 + sz/2, y + sz/2);
        draw(n-1, sz/2, x0, y0);
        draw(n-1, sz/2, x0, y1);
        draw(n-1, sz/2, x1, y0);
        draw(n-1, sz/2, x1, y1);
    }
    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        draw(n, .5, .5, .5);
    }
}
```
Animated H-tree

Animated H-tree. Pause for 1 second after drawing each H.

Fractional Brownian Motion

Physical process which models many natural and artificial phenomenon.
- Price of stocks.
- Dispersion of ink flowing in water.
- Rugged shapes of mountains and clouds.
- Fractal landscapes and textures for computer graphics.

Simulating Brownian Motion

Midpoint displacement method.
- Maintain an interval with endpoints \((x_0, y_0)\) and \((x_1, y_1)\).
- Divide the interval in half.
- Choose \(\delta\) at random from Gaussian distribution.
- Set \(x_m = (x_0 + x_1)/2\) and \(y_m = (y_0 + y_1)/2 + \delta\).
- Recur on the left and right intervals.

Simulating Brownian Motion: Java Implementation

Midpoint displacement method.
- Maintain an interval with endpoints \((x_0, y_0)\) and \((x_1, y_1)\).
- Divide the interval in half.
- Choose \(\delta\) at random from Gaussian distribution.
- Set \(x_m = (x_0 + x_1)/2\) and \(y_m = (y_0 + y_1)/2 + \delta\).
- Recur on the left and right intervals.

public static void curve(double x0, double y0, double x1, double y1, double var) {
    if (x1 - x0 < 0.01) {
        StdDraw.line(x0, y0, x1, y1);
        return;
    }
    double xm = (x0 + x1) / 2;
    double ym = (y0 + y1) / 2;
    ym += StdRandom.gaussian(0, Math.sqrt(var));
    curve(x0, y0, xm, ym, var/2);
    curve(xm, ym, x1, y1, var/2);
    // variance halves at each level; change factor to get different shapes
}

Plasma Cloud

Plasma cloud centered at \((x, y)\) of size \(s\).
- Each corner labeled with some grayscale value.
- Divide square into four quadrants.
- The grayscale of each new corner is the average of others.
- - center: average of the four corners + random displacement
- - others: average of two original corners
- Recur on the four quadrants.
Plasma Cloud

Brownian Landscape

Robert Brown (1773-1858)