CSC 1052 – Algorithms & Data Structures II:

Recursive Applications

Professor Henry Carter
Spring 2017

Recap

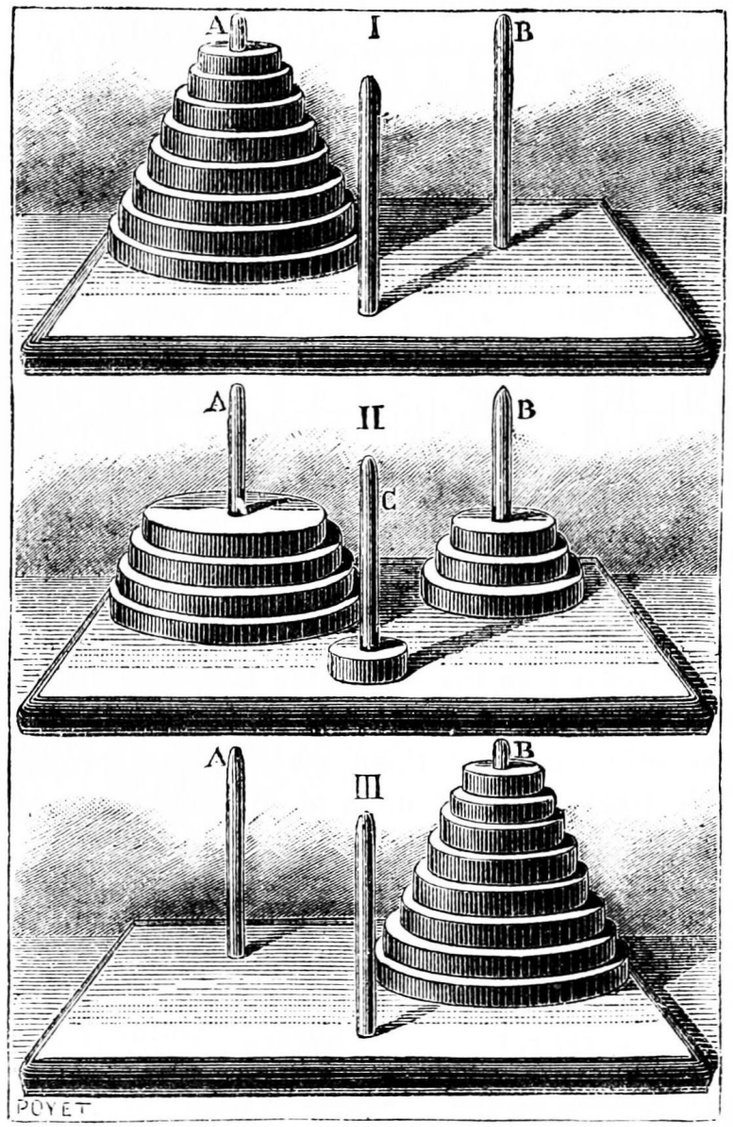
- Data structures may have recursive structure
- Recursively processing linked lists allows for simplified code that takes advantage of this structure
- Recursion makes some tasks easier but may lead to unexpected pitfalls
  - Remember, Java is call by VALUE

Example Applications

- Some obvious use cases
  - Factorial
- Some less intuitive solutions
  - Towers of Hanoi
- Some intrinsic value
  - Generating recursive structure (fractals)

Towers of Hanoi

- Legend
- Setting
  - Three pegs
  - Pyramid of rings
  - Goal: move the pyramid to the far peg
- Rules
  - One ring moved at a time
  - A larger ring can never sit atop a smaller ring



4-Ring Example

N-Ring Technique

- Recursive solution takes advantage of the auxiliary pole
- Base case:
  - 1 ring, 1 move
- Recursive case
  - Move n-1 rings to the aux pole
  - Move the biggest ring to the end pole
  - Move n-1 rings to the end pole

Implementation in Java

- Follow along in your own editor

Three Questions

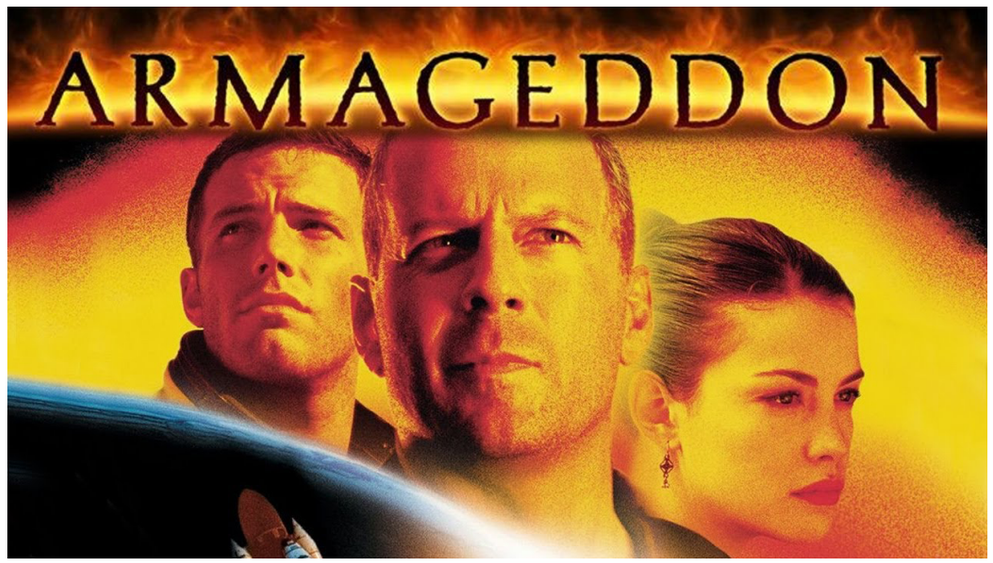
- Base case

- Smaller-caller

- General case

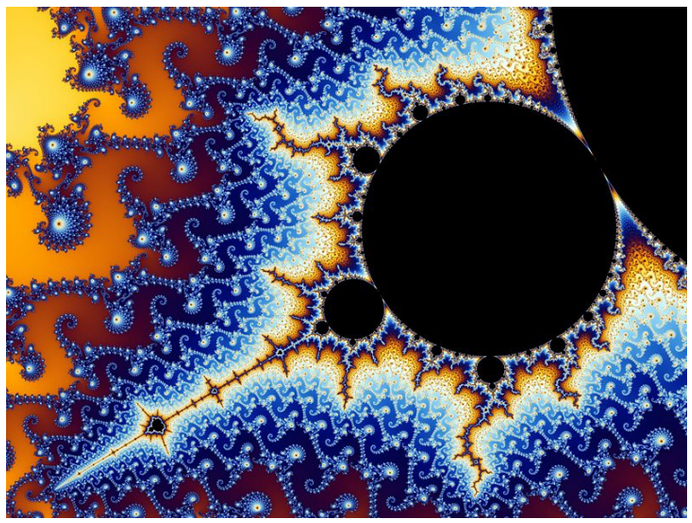
Analysis

- Number of moves doubles with each call
- Mathematical analysis: $2^n - 1$
- Original legend described 64 rings



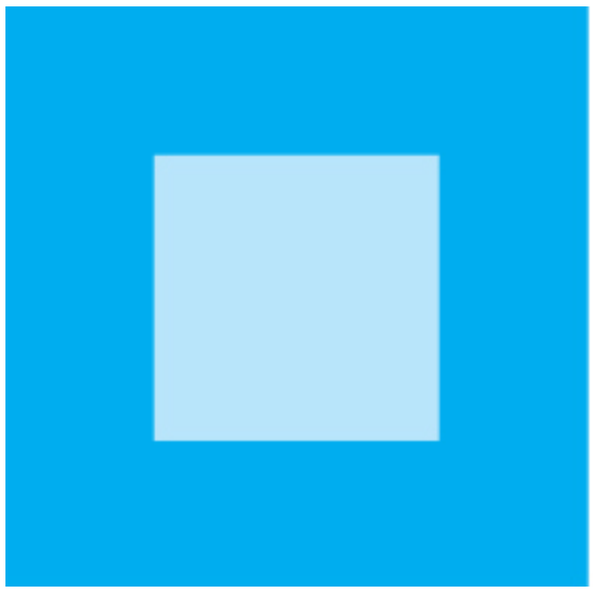
Fractals

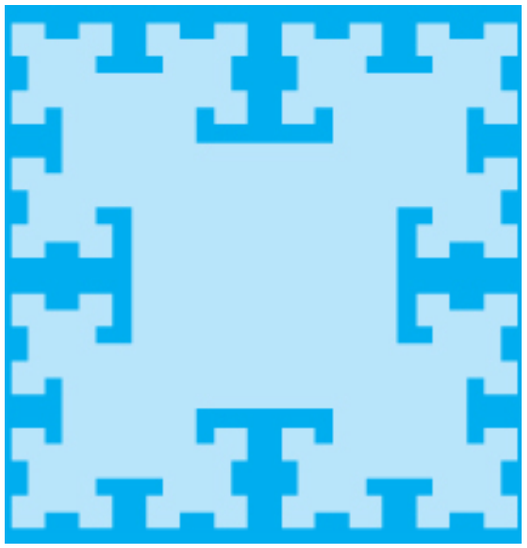
- Repeated structure within a structure
- In images: a smaller version of a picture inside the picture
- Appearance in the real world:
  - Natural structures
  - Strong engineering and architecture
  - Mathematical and graphical modelling

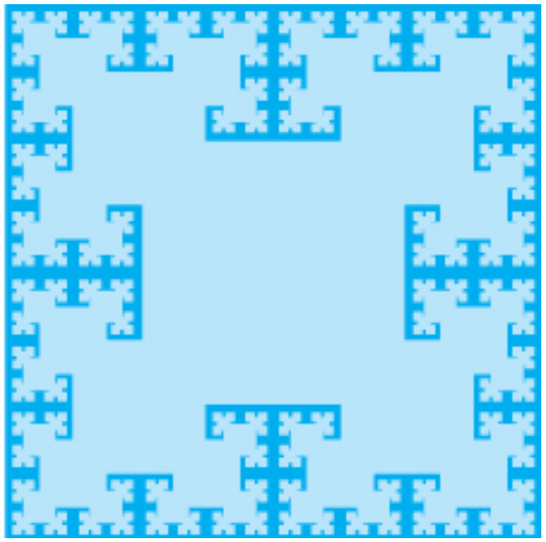


T-Square Fractals

- Draw a white square on a canvas
- Recursively repeat with each canvas divided into four







Java Images

- A simple representation of an image is a 2D array of sRGB values
- Shading and color can be adjusted from 0 to 255 (??-bit color?)
- The alpha component determines transparency
- We use the awt.image package

T-Square: Setup Code

```
public class TSquare
{
   static final int SIDE = 1000;    // image is SIDE X SIDE
   static BufferedImage image = new BufferedImage(SIDE, SIDE, BufferedImage.TYPE_INT_RGB);
   static final int WHITE = Color.WHITE.getRGB();
   static final int BLACK = Color.BLACK.getRGB();

                                    ....

  public static void main (String[] args) throws IOException
  {
   String fileOut = args[0];

   // make image black
   for (int i = 0; i < SIDE; i++)
     for (int j = 0; j < SIDE; j++)
     {
        image.setRGB(i, j, BLACK);
     }

   // first square
   drawSquare(SIDE/2, SIDE/2, SIDE/2);

   // save image
   File outputfile = new File(fileOut);
   ImageIO.write(image, "jpg", outputfile);
}
```

T-Square: Recursive Work

```
private static void drawSquare(int x, int y, int s)
// center of square is x,y  length of side is s
{
   if (s <= 0)  // base case
      return;
   else
   {
      // determine corners
int left = x - s/2;
int top  = y - s/2;
int right = x + s/2;
int bottom = y + s/2;

// paint the white square
for (int i = left; i < right; i++)
  for (int j = top; j < bottom; j++)
  {
     image.setRGB(i, j, WHITE);
  }
```

T-Square: Recursive Call

```
// recursively paint squares at the corners
drawSquare(left, top, s/2);
drawSquare(left, bottom, s/2);
drawSquare(right, top, s/2);
drawSquare(right, bottom, s/2);
```

Exercise: make your own!

- Today's lab will allow you to experiment with recursive fractals
- Create another fractal: Sierpinski carpet
- Try some variations from the book
- Try your own variations

Recap

- Recursion is an obvious solution for many mathematical problems
- Recursion may be able to solve some problems in more subtle ways
  - Towers of Hanoi
- Recursion can be used as a tool on its own
  - Generating recursive structures in fractals

Next Time...

- Dale, Joyce, Weems Chapter 3.7-3.8
  - Remember, you need to read it BEFORE you come to class!
- Check the course webpage for practice problems
- Peer Tutors
  - http://www.csc.villanova.edu/help/