BST 140.778 Assignment 1

January 22, 2005

Fine print All computing assignments must be completed in the R statistical programming language. For non-computing assignments please appropriately typeset using LaTeX. Bundle your R functions and LaTeX files for each assignment in a `zip` or `tar.gz` file and email them to `bcarvalh@jhsph.edu`. Include *specific* instructions on how to run the code to answer the assignments in a `README` file. All code should be readable, formatted well, commented and clearly indicate the author and date. The general rule is: the more thought that has to go into understanding how to implement your code, the worse your grade will be. Please feel free to give each other small hints, but otherwise students must complete assignments individually.

Assignment 1 This assignment is to write a few R programs to get familiar with the language.

1. Write an R function `powerset` that takes a vector and produces a list of the power set of the elements of that vector. Recall the power set of a vector of elements is the set of all possible sets with elements from that vector. So, if the vector has n elements the returned list should have 2^n elements.

   ```
   Function: powerset
   Inputs  : vector
   Ouput   : list where each list element is one vector
             element of the power set
   Usage   : powerset(vector)
   Example : powerset(letters[1 : 5])
   ```

2. Write a recursive R function `blocks` that produces Figure 1 for a specific number of iterations. Shown is, 1, 2, 3 and 4 iterations.

   ```
   Function: blocks
   Inputs  : n
   Ouput   : plot after n iterations
   Usage   : blocks(n)
   ```

3. Write an R function `kaprekar` to perform Kaprekar's process. This process takes an initial non negative 3 digit integer with unique digits, like 394, then subtracts the integer obtained by ordering the digits from least to greatest by the integer obtained by ordering the digits from greatest to least, 943 - 349 $=$ 594. Then the process is repeated on the result. The sequence will eventually terminate at 495. This is an example of an iterated function. That is, if f takes an integer and performs one Kaprekar iteration then the sequence x $f(x)$, $f(f(x))$, ... will terminate at 495. Note that $f(495) = 495$, so that 495 is a fixed point of f. The fixed point theorem, which states that many functions that are repeatedly applied to a starting point converge to a fixed point, is a theorem that underlines much of what we will be doing in this course.

```
Function: kaprekar
Inputs  : n a three digit integer with
          three unique digits
Outputs : the number of iterations for the
          kaprekar process to reach 495 starting
          at n
```

4. The *Greatest Common Divisor* (gcd) of two integers, say a and b, is the largest common divisor of both a and b. For example, GCD$(15, 35) = 5$. Euclid's algorithm, one of the oldest formal algorithms in existence (written in 300 BC), calculates the gcd of two numbers. The algorithm proceeds as follows

   Divide a by b and keep the remainder, r_1

   Divide b by r_1 and keep the remainder, r_2

   Divide r_2 by r_1 and keep the remainder, r_3

   $\vdots$

   Eventually, there will occur an i so that $r_i = 0$. The GCD of a and b is then r_{i-1}. Write an R function gcd to calculate the greatest common divisor of two integers using Euclid's algorithm.

```
Function: gcd
Inputs  : integers a and b
outputs : gcd of a and b using Euclid's algorithm
```

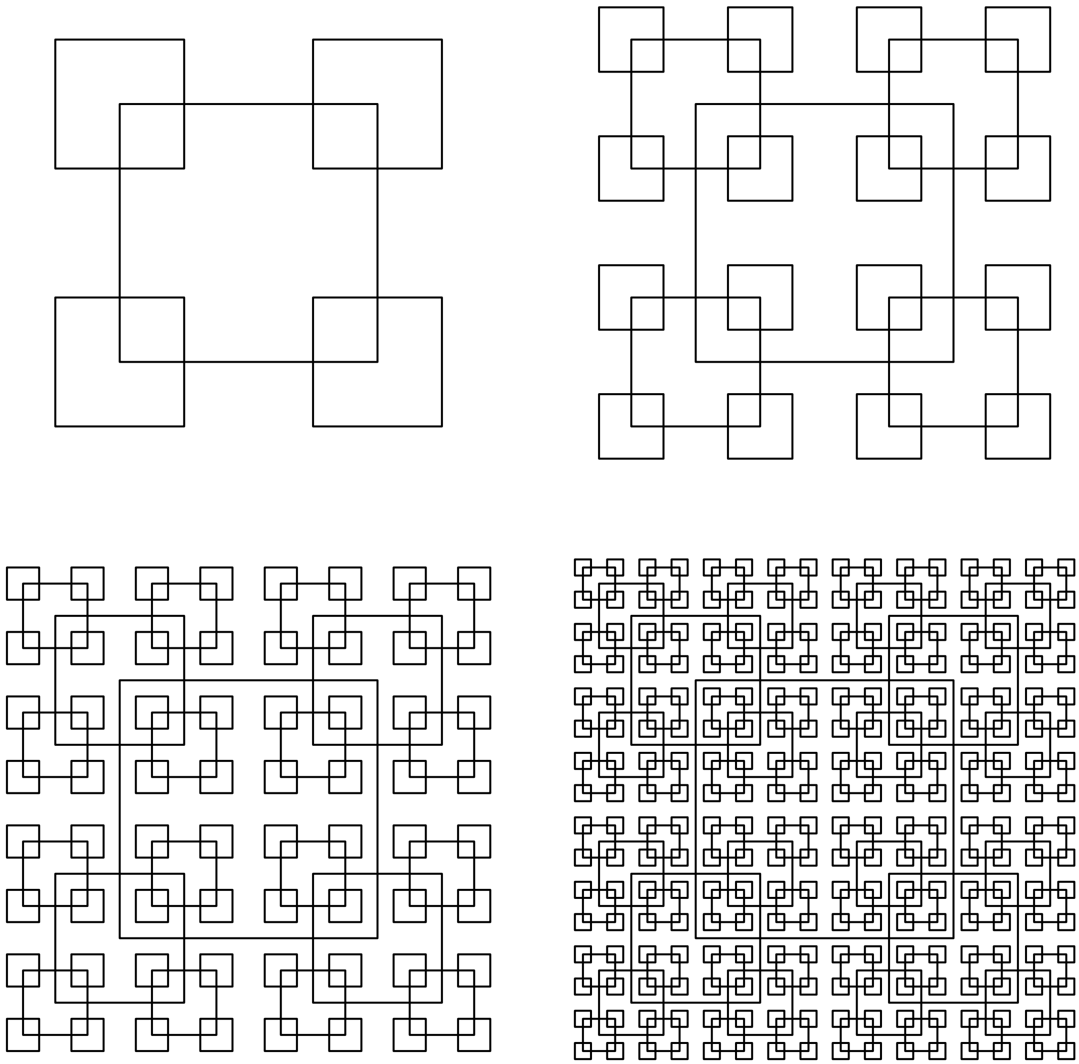

Figure 1: Iterations 1, 2, 3, and 4 of `blocks`