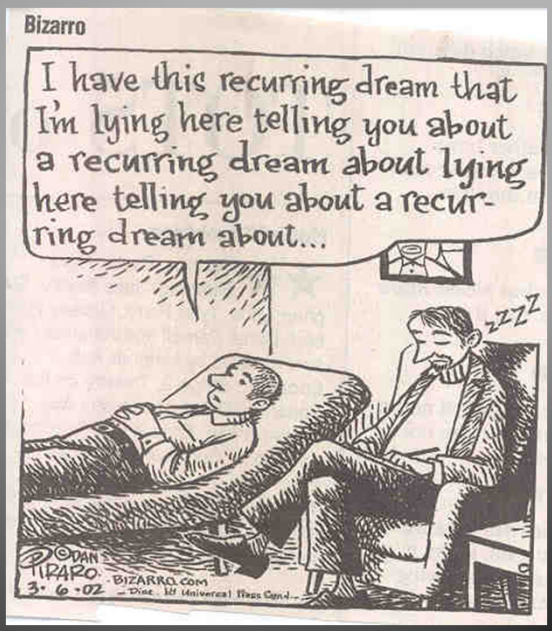
CSSE 220 Day 12

Sierpiński, Recursion and Efficiency, Mutual Recursion

Checkout *Recursion2* project from SVN



Exam 1

- Format:
  - Paper Part
    - Emphasis: ideas (syntax errors will be considered minor)
  - Computer Part
    - Emphasis: Actually getting code to work (almost all points will be for passing unit tests).
  - You can decide how much time to spend on each part, but you must turn in the Paper Part before you use your computer.

Resources for Paper Part

- Textbook
- Notes sheet:
  - Single 8.5 by 11 page (both sides)
  - Whatever you want on it
  - Prepare this carefully!

No Sharing !

Resources for Computer Part

- Any printed or handwritten material you choose (notes, books, printouts, …)
- Your computer, with power adapter and network cable
  - Your computer and anything actually on it
  - Network, but only to access your own SVN repository and any material directly reachable from the CSSE 220 ANGEL and web sites for this term
  - No chat programs, email, or other means of communication.
- No cell phones or devices with earphones.

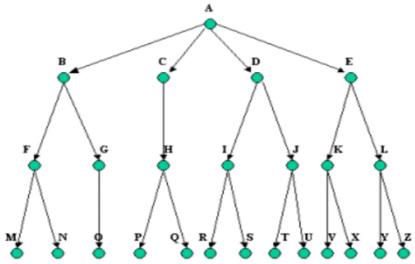
Questions?

What are recursive methods?

- Any method that calls itself
  - On a simpler problem
  - So that it makes progress toward completion

When should recursive methods be used?

- When implementing a recursive definition
- When implementing methods on recursive data structures



- Where parts of the whole look like smaller versions of the whole

Q1

The pros and cons of recursive methods

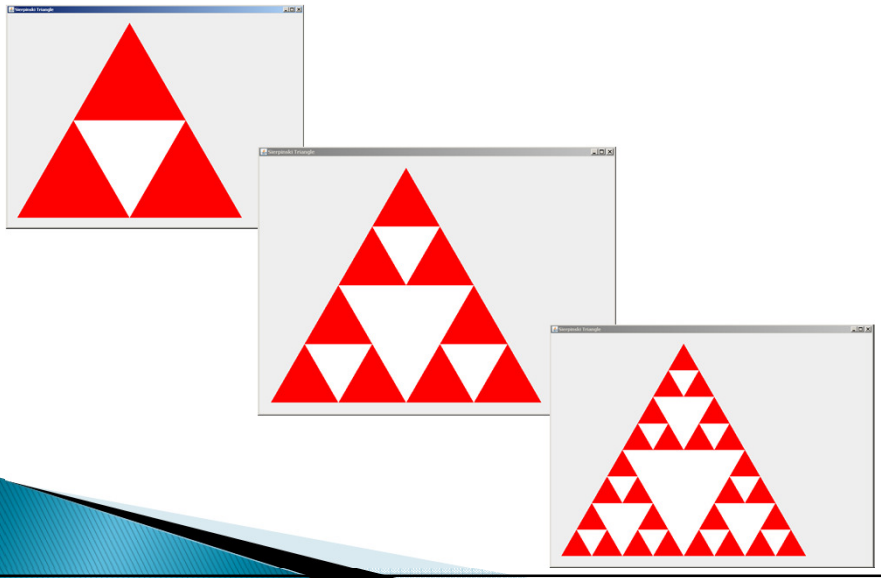
- The pros
  - easy to implement,
  - easy to understand code,
  - easy to prove code correct
- The cons
  - takes more space and time than equivalent iteration
  - Why?
    - because of function calls

Q2

Recap: Key Rules to Using Recursion

- Always have a **base case** that **doesn't recurse**
- Make sure recursive case always makes **progress**, by **solving a smaller problem**
- **You gotta believe**
  - Trust in the recursive solution
  - Just consider one step at a time

HW: Sierpinski



Work Time

- HW 12 & 13: Sierpinski Triangle

Can one little Fib hurt?

- Why does recursive Fibonacci take so long?!?
- Can we fix it?

```
private static long fib(int n) {
    // TODO: Convert this to use memoization.
    long f;
    if (n <= 2) {
        f = 1;
    } else {
        long fNMOne = fib(n - 1);
        long fNMTwo = fib(n - 2);
        f = fNMOne + fNMTwo;
    }
    return f;
}
```

Q3

Memoization

- Save every solution we find to sub-problems
- Before recursively computing a solution:
  - Look it up
  - If found, use it
  - Otherwise do the recursive computation

Q4

Classic Time-Space Trade Off

- A deep discovery of computer science
- In a wide variety of problems we can tune the solution by varying the amount of storage space used and the amount of computation performed
- Studied by "Complexity Theorists"
- Used everyday by software engineers

Q5

Mutual Recursion

- 2 or more methods call each other repeatedly
  - E.g., Hofstadter Female and Male Sequences

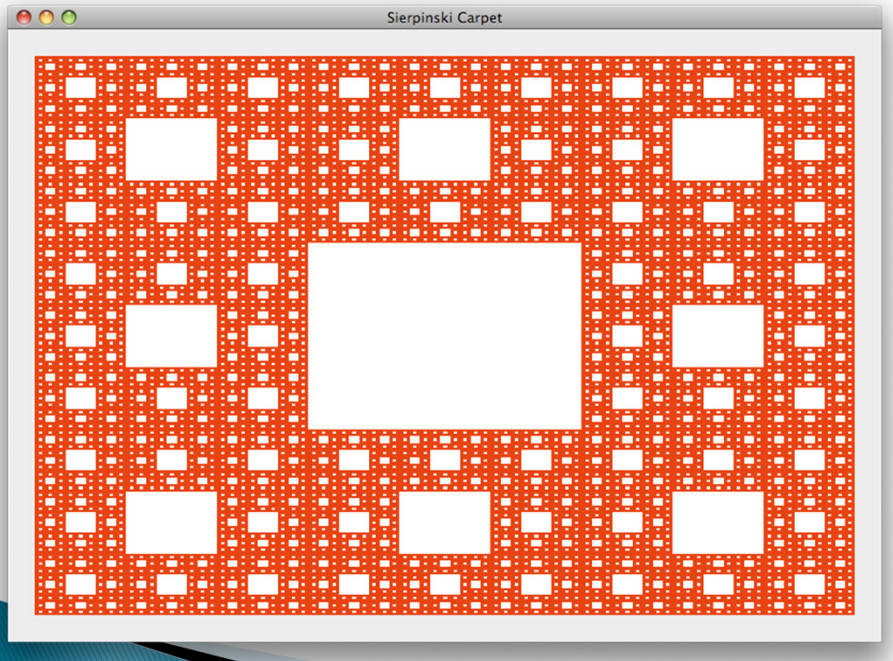
$$F(n) = \begin{cases} 1 & \text{if } n = 0 \\ n - M(F(n-1)) & \text{if } n > 0 \end{cases}$$

$$M(n) = \begin{cases} 0 & \text{if } n = 0 \\ n - F(M(n-1)) & \text{if } n > 0 \end{cases}$$

  - In how many positions do the sequences differ among the first 50 positions? first 500? first 5,000? first 5,000,000?

http://en.wikipedia.org/wiki/Hofstadter_sequence

Q6




Work Time
HW 12: Sierpinski Carpet
Due Monday
Q7-8