More Recursion!

Recursion - examples
- Problem: Given a string as input, write it backward
- Base case?
- Recursion

Tail recursion
- Tail recursion is a recursive call that occurs as the last action in a method.
- This is not tail recursion:
  ```java
  public int factorial(int n) {
    if (n == 0)
      return 1;
    return n * factorial(n-1);
  }
  ```

Tail recursion
- This is tail recursion:
  ```java
  public int factorial(int n) {
    return factorialTail(n, 1);
  }
  ```

Tail recursion
- But why would you care? Turns out that compilers can optimize memory usage when they detect that this is the case.

Tail recursion
- This is tail recursion:
  ```java
  public int factorial(int n) {
    return factorialTail(n, 1);
  }
  ```

Tail recursion
- This is tail recursion:
  ```java
  public int factorial(int n) {
    return factorialTail(n, 1);
  }
  ```

- When making a recursive call, you no longer need to save the information about the local variables within the calling method.
Dictionary lookup

- Suppose you’re looking up a word in the dictionary (paper one, not online!)
- You probably won’t scan linearly thru the pages — inefficient.
- What would be your strategy?

Binary search

- Let’s write a method called `binarySearch` that accepts a sorted array of integers and a target integer and returns the index of an occurrence of that value in the array.
- If the target value is not found, return -1

| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Value | -4| 2 | 7 | 10| 15| 20| 22| 25| 30| 36| 42| 50| 56| 85| 92 | 103|

```
int index = binarySearch(data, 42); // 10
int index2 = binarySearch(data, 66); // -1
```

Towers of Hanoi

Example: Towers of Hanoi, move all disks to third peg without ever placing a larger disk on a smaller one.

Try to find the pattern by cases

- One disk is easy
- Two disks...
- Three disks...
- Four disk...
Towers of Hanoi

Example: Towers of Hanoi, move all disks to third peg without ever placing a larger disk on a smaller one.

Let's go play with it at: http://www.mazeworks.com/hanoi/index.htm
Or http://www.mathsisfun.com/games/towerofhanoi.html

Fibonacci’s Rabbits

- Suppose a newly-born pair of rabbits, one male, one female, are put on an island.
  - A pair of rabbits doesn’t breed until 2 months old.
  - Thereafter each pair produces another pair each month.
  - Rabbits never die.
- How many pairs will there be after \( n \) months?

Fibonacci numbers

- The Fibonacci numbers are a sequence of numbers \( F_0, F_1, \ldots, F_n \) defined by:

\[
F_0 = F_1 = 1 \\
F_i = F_{i-1} + F_{i-2} \text{ for any } i > 1
\]

- Write a method that, when given an integer \( i \), computes the \( n \)th Fibonacci number.
Fibonacci numbers

- Let's run it for \( n = 1, 2, 3, \ldots, 10, \ldots, 20, \ldots \).
- If \( n \) is large the computation takes a long time! Why?

If \( n \) is large the computation takes a long time! Why?

Fibonacci numbers

- recursive Fibonacci was expensive because it made many, recursive calls
  - \( \text{fibonacci}(n) \) recomputed \( \text{fibonacci}(n-1), \ldots, \text{fibonacci}(1) \) many times in finding its answer!
  - this is a case, where the sub-tasks handled by the recursion are redundant with each other and get recomputed

Fibonacci numbers

- Every time \( n \) is incremented by 2, the call tree more than doubles.

Growth of rabbit population

1 1 2 3 5 8 13 21 34 ...

The fibonacci numbers themselves also grow rapidly: every 2 months the population at least **DOUBLES**

Fractals – the Koch curve and Sierpinski Triangle