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);  
}
int factorialTail(int n, int product) {
    if(n == 0)
        return product;
    return factorialTail(n-1, product*n);  
}
```
Tail recursion

- This is tail recursion:
  ```java
  public int factorial(int n) {
    return factorialTail(n, 1);
  }
  int factorialTail(int n, int product) {
    if(n == 0)
      return product;
    return factorialTail(n-1, product*n);
  }
  
  But why would you care? Turns out that compilers can optimize memory usage when they detect that this is the case.
  ```

- 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

```java
binarySearch(dictionary, word){
  if {dictionary has one page} // base case
    scan the page for word
  else // recursive case
    open the dictionary to a point near the middle
    determine which half of the dictionary contains word
    if (word is in first half of the dictionary) {
      binarySearch(first half of dictionary, word)
    } else {
      binarySearch(second half of dictionary, word)
    }
}
```
### 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

```java
int index0 = 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...
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:
  \[
  F_0 = F_1 = 1
  \]
  \[
  F_i = F_{i-1} + F_{i-2} \quad \text{for any } i > 1
  \]
- Write a method that, when given an integer \( i \), computes the \( n \)th Fibonacci number.
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