Recursion

- Simple Example
- Definition
- Standard Examples

Repeating Patterns

- base recursive

A square is either a 1x1 box, or is made of 4 smaller squares such that the width = height

Definition

- A structure is recursive iff it can be expressed in terms of smaller versions, or it is the smallest version already.
- These are called recursive case and base case (smallest).
Factorial

- $n! = 1 \times 2 \times ... \times n$
- $1! = 1$
- $2! = 1 \times 2$
- $3! = 2! \times 3$
- $4! = 3! \times 4$
- $n! = (n-1)! \times n$ or if $n = 1$, $n! = 1$

Implementation

```java
public static int factorial(int n) {
    if (n == 0) {
        return 1;
    } else {
        return factorial(n - 1) * n;
    }
}
```

```
/**
 * Calculate factorial of n.
 * pre: n not negative
 * post: n! is returned.
 * @param n integer number to compute fact of.
 * @return integer n!
 */
public static int factorial(int n) {
    if (n == 0) {
        return 1;
    } else {
        return factorial(n - 1) * n;
    }
}
```

Strategy

- Define problem in terms of smaller versions of itself
- Call function from itself using SMALLER versions
- Determine the BASE case
- Make progress towards BASE case

Sum(n) = ?

- $\sum(n) = 1 + 2 + \ldots + n; \sum(1) = 1$
- $\sum(n) = \sum(n-1) + n; \sum(1) = 1$
How does it work?

- stack of activation records
- function call = push
- function return = pop
- parameters and local variables stored in an activation record

Problems

- Print string backwards
- Print a list w/o current, position
- Binary Search
- Towers of Hanoi