

CSSE 220

Recursion

Checkout *Recursion* project from SVN

Recursion

- A solution technique where the same computation **occurs repeatedly** as the problem is solved



recurs

- Examples:
 - Sierpinski Triangle
 - Towers of Hanoi:
<http://www.mathsisfun.com/games/towerofhanoi.html>
or search for Towers of Hanoi

An example – Triangle Numbers

- If each red block has area 1, what is the **area $A(n)$** of the Triangle whose *width* is n ?

– Answer:

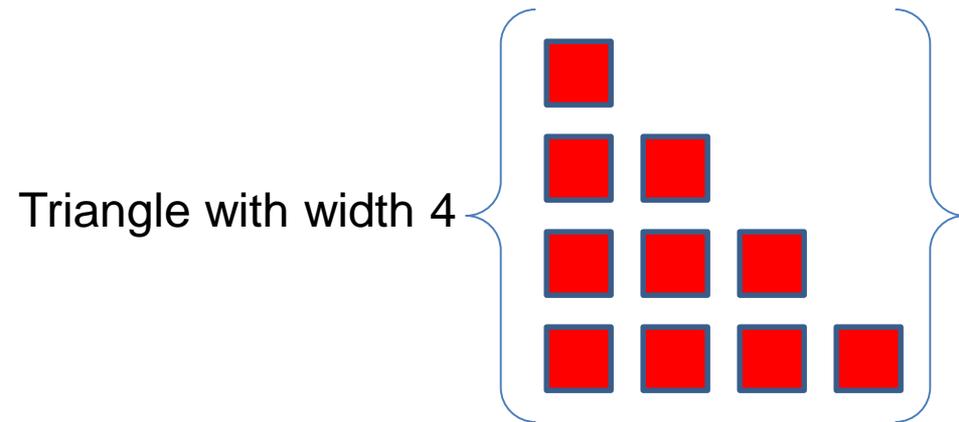
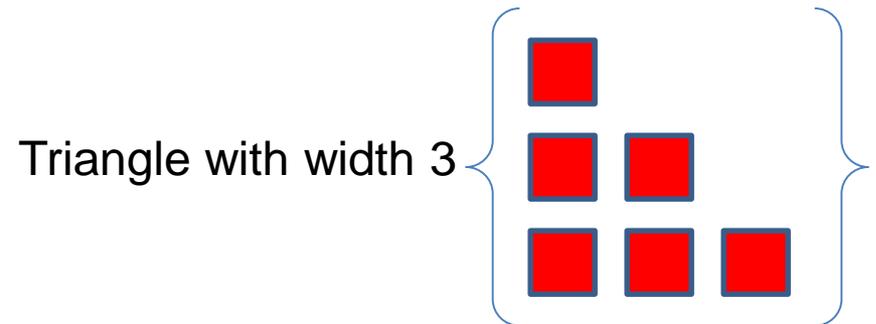
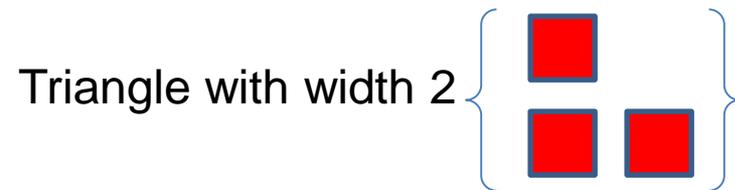
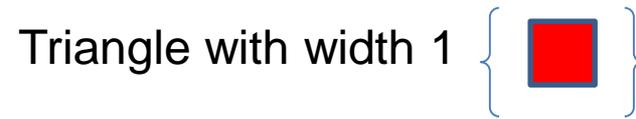
$$A(n) = n + A(n-1)$$

- The above holds for which n ? What is the answer for other n ?

– Answer: The recursive equation holds for

$$n \geq 1.$$

For $n = 0$, the area is 0.



Frames for Tracing Recursive Code

1. Draw box when method starts

2. Fill in name

method name

parameters

local variables

3. List every parameter and its argument value.

4. List every local variable declared in the method, **but no values yet**

6. Step through the method, update the line number and variable values, draw new frame for new calls

7. "Erase" the frame when the method is done.

Q1-Q2

Thanks to David Gries for this technique

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

Programming Problem

- Add a recursive method to Sentence for computing whether Sentence is a palindrome

Sentence
String text
String toString() boolean isPalindrome

Recursive Helpers

- ▶ Our `isPalindrome()` makes lots of new Sentence objects
- ▶ We can make it better with a “recursive helper method”
 - ▶ Many recursive problems require a helper method

```
public boolean isPalindrome() {  
    return isPalindrome(0, this.text.length() - 1);  
}
```

Position of first letter of the remaining String to check

Position of last letter of the remaining String to check

Practice Practice Practice

- Head to <http://codingbat.com/java/Recursion-1> and solve 5 problems. I personally like bunnyEars, bunnyEars2, count7, fibonacci, and noX
- Get help from me if you get stuck
- Then take a look at the RecursionPractice assignment (due Wednesday after break)