**Recursion**

* Recursion is a problem-solving approach that can be used to generate simple solutions to certain kinds of problems that would be difficult to solve in other ways
* Recursion splits a problem into one or more simpler versions of itself

**Steps to Design a Recursive Algorithm**

* There must be at least one case (the base case), for a small value of n, that can be solved directly
* A problem of a given size n can be split into one or more smaller versions of the same problem (recursive case)
* Recognize the base case and provide a solution to it
* Devise a strategy to split the problem into smaller versions of itself while making progress toward the base case
* Combine the solutions of the smaller problems in such a way as to solve the larger problem

**Recursive Algorithm for finding the length of a String**

1. If the string is empty
2. the length is 0  
   else
3. The length is 1 plus the length of the string that excludes the first character

**Proving that a Recursive Method is Correct**

**Proof by induction**

* Prove the theorem is true for the base case
* Show that if the theorem is assumed true for n, then it must be true for n+1

**Recursive proof is similar to induction**

* Verify the base case is recognized and solved correctly
* Verify that each recursive case makes progress towards the base case
* Verify that if all smaller problems are solved correctly, then the original problem is also solved correctly

**General Fact**

Anything that can be solved iteratively, can be solved recursively!

* Infinite Recursion

**Advantages of Recursion**

* Often, the code is easier to write
* Often, the code is easier to understand
* Recursion is a more powerful problem solving technique than iteration

**Advantages of Iteration**

* There is no method call overhead
* Sometimes, the natural iterative solution has a better time complexity than the natural recursive solution

**Code from class:**

* [Example](https://www.cs.montana.edu/hunterl/221/lectures/recursion/MethodCall.java)showing the stack without recursion
* [First Recursion example](https://www.cs.montana.edu/hunterl/221/lectures/recursion/RecMethodCall.java), I will put the activation records on the board that displayed the Stack.
* [Print the chars example](https://www.cs.montana.edu/hunterl/221/lectures/recursion/PrintChars.java), and printing in reverse example
* [The examples from the book](https://www.cs.montana.edu/hunterl/221/lectures/recursion/RecursiveMethods.java)