Program 1 Grades available – If you have questions, contact your grader by 11/1
Program 3 assigned – teams by 11/1 – due by 10pm on Sunday November 6th, -- don't wait!
Homework 6 quiz – available Wed 10/26 10am, due by 10pm Friday October 28th

Last Time
  PriorityQueue Review
  Java’s Comparable Interface
  Heap Data Structure
  • insert
  • removeMax
  Java’s Stack, Queues, PriorityQueues
  Call Stack Tracing

Today
  Java’s Stack, Queues, PriorityQueues (from last time)
  Call Stack Tracing (from last time)
  Recursion
  • recursion vs. iteration
  • rules of recursion
  • constructing recursive code
  • practice writing recursive code
  Exam 1 returned

Next Time
  Read: continue Recursion
  Recursion
  ✴ more practice writing recursive code
  ✴ complexity of recursive methods
  ✴ practice analyzing complexity
Recursion vs. Iteration

Recursion is like iteration:

| Iteration | Recursion |

Recursion is NOT like iteration:

- Each loop iteration
- A loop with a bad stopping condition

Rules for Recursion

1. ...

2. ...
Recursion

What is it?

Why use it?

→ How would you modify the print method to display a singly-linked chain of nodes in reverse order?
Factorials: $n!$

Consider the factorial of $n$ (assume $n \geq 0$):

$$n! = n \times (n-1) \times (n-2) \times (n-3) \times \ldots \times 2 \times 1$$

$6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$

Method Call:

```java
factorial(6);
```

Iterative Implementation:

```java
int factorial(int n) {
    int result = 1;
    for (; n > 1; n--)
        result = result * n;
    return result;
}
```

Recursive Definition:

→ Complete the Recursive Implementation:

```java
int factorial(int n) {
```
Constructing Recursive Code

→ Write a recursive method that computes \( n^m \)
that is, it computes double \( n \) raised to an int power \( m \)?

recursive definition:

recursive implementation:

Key Questions:
1.
2.
3.
4.
Write a recursive method that displays the values in a (non-null) list of strings.

```java
void display(ListADT<String> list) {
```

Copyright 2014-2016 Jim Skrentny
Making a Reversed Copy of a Chain of Nodes

Method Use:
```java
Listnode<String> reversed = reverseCopy(head, null);
```

Method Implementation:
```java
Listnode<String> reverseCopy
    (Listnode<String> curr, Listnode<String> rev) {
        if (curr == null) return rev;
        rev = new Listnode<String>(curr.getData(), rev);
        return reverseCopy(curr.getNext(), rev);
    }
```

How does it work?

![Diagram of a reversed copy of a chain of nodes](image)