**Up-Down sequence**

An up-down sequence of length n is a reordering of the numbers 1, 2, 3, ..., n such that the second number is larger than the first, the third is smaller than the second, the fourth is larger than the third, and so on in an alternating "up-down" way.

Here are all the up-down sequences of length 4 or less:

n=1: <1>  
n=2: <1,2>  
n=3: <1,3,2> <2,3,1>  
n=4: <1,3,2,4> <1,4,2,3> <2,3,1,4> <2,4,1,3> <3,4,1,2>

You are to find the total number of "up-down sequences" of length n, for 1 <= n <= 10.

It turns out that a triangle of numbers (shown below) is a way to solve this problem:

1

0 -> 1

1 <- 1 <- 0

0 -> 1 -> 2 -> 2

5 <- 5 <- 4 <- 2 <- 0

0 -> 5 -> 10 -> 14 -> 16 -> 16

The triangle is formed by starting with 1 on top. Row 2 is formed from left to right by starting with 0 and then adding the numbers in the row above. Row 3 is formed from right to left starting with 0 and then adding the numbers in the row above. The directions alternate as you proceed (follow the arrows in the diagram). Row 5, for example, goes from right to left and starts with 0 (on the right). Then add the numbers in row 4 (the previous row) as follows: 0 + 2 = 2, 2 + 2 = 4, 4 + 1 = 5, 5 + 0 = 5. You should get:

61 <- 61 <- 56 <- 46 <- 32 <- 16 <- 0

…when you try to generate row 6 by hand.

The number in column k of row n of the triangle is the number of up-down sequences of length n that end in k (the column number). For example, row 4 indicates the numbers of up-down sequences of length 4. There are 0 ending with the number 1, 1 ending with 2, 2 ending with 3 and 2 ending with 4. See the list above for verification. This gives a total of 0 + 1 + 2 + 2 = 5 sequences of length 4. The total number of up-down sequences of length n is the sum of entries in row n.

*Credit: The idea for this lab came from a programming problem used at Denver University in 1997.*

**The Assignment**

This lab will require a ***doubly*** linked list. Previously, you wrote a singly linked list, where each node stored only a reference to the next node. Doubly linked lists store a reference to the previous *and* next nodes, such that the list can be traversed in either direction.

Conveniently, the java.util.LinkedList data structure's underlying implementation uses a doubly-linked list! In addition to the standard List methods, LinkedList supplies methods to quickly get, insert, and remove elements at the beginning and end of the list. The algorithm described above works well using removeLast(), removeFirst(), addLast(), and addFirst() methods appropriately.

Your solution should generate the total number of up-down sequences for every *n* from 1 to 10 and display it in a list in this format:

length = 2, sequences >>> 1  
length = 3, sequences >>> 2  
length = 4, sequences >>> 5  
length = 5 ...

This project based on the Up-Down lab

*http://www.rfrank.net/cslabs/1880-updown/1880.html*