Simple Recursion

Due: Day of lab at 11:59PM

1 Simple Recursion Specification

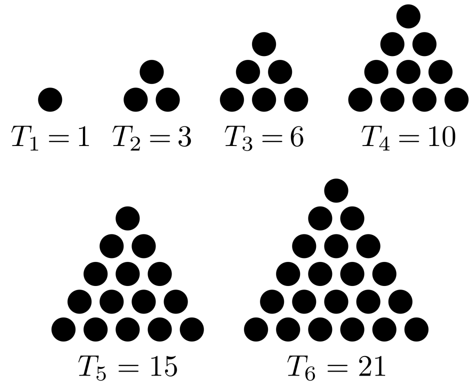
1.1 Lab Link

The skeleton code for the lab is available at `https://classroom.github.com/a/Fc_LJ8Ca`.

1.2 Implementation

For this lab, you will implement and test a simple recursive method that calculates triangular numbers.

Triangular numbers represent something similar to square numbers. For n^2, you count how many dots there are in a $n \times n$ square. For a triangular number T_n, you count how many dots there are in an equilateral triangle with sides of n. Negative values of n are not allowed.



First six triangular numbers (after the empty triangle)

The function T_n can be generally described as:

$$T_n = \begin{cases} 0, & \text{if } n = 0 \\ n + T_{n-1}, & \text{otherwise} \end{cases}$$

In the **Triangular** class, implement a *recursive* method to count the dots in a triangle. The supplied method is a wrapper method, so you should implement a private method as well, similar to the class examples.

1.3 Testing

Testing should be straightforward, as solutions are easy to anticipate. Implement the **testValid** JUnit test to ensure that you are calculating correct answers.

1.4 Available Resources

- Lecture slides
- Other sections of the provided code
- me
- The textbook
- **DO NOT** refer to or use online implementations

1.5 Lab Instructions

- This is an *individual* lab.
- Make sure to read through all of the specifications so your submission is complete.
- Follow all the submission steps in the Setup document by the lab deadline.

2 Double Check:

- Have you implemented the `countTriangle` method?
- Have you written a JUnit test?
- Have you committed/pushed your code from the two files?

3 Grading

Each of the 2 **TODO** sections is worth $\frac{1}{2}$ of the lab grade.

Grades and any comments for the lab will be posted to your project on github. Grades will also be posted to Brightspace, eventually.