CSCI 239—Discrete Structures of Computer Science
Lab 8—Haskell Functions 2

In this lab, you will write and test Haskell functions that implement recurrence relations; in the next lab you'll use these Haskell functions and other tools to solve these recurrence relations.

Objectives:

- to gain experience writing recursive Haskell functions
- to use Haskell functions as an aid to solving recurrence relations.

Create a *Lab08* folder in your *CS239* folder and put all code you write in this lab into this folder.

For each of the following recurrence relations, write a Haskell function implementing the recurrence relation. Then use the *map* function to compute the first ten values for the function. Record these values in a comment after the code for each function. Your function does not need to handle values the recurrence relation is not defined for.

For example, if your function is *relA* and the base value is zero, you can use:

```
>> map relA [0 .. 9]
```

to compute `relA n` for the values zero to nine.

Here are the recurrence relations to code in Haskell:

$$F(n) = \begin{cases} 5 & n = 1 \\ F(n-1) + 3n - 2 & n > 1 \end{cases}$$

$$G(n) = \begin{cases} 3 & n = 0 \\ G(n-1) + 2n + 4 & n > 0 \end{cases}$$

$$H(n) = \begin{cases} 2 & n = 0 \\ H(n-1) + 3n^2 - 4n + 1 & n > 0 \end{cases}$$

$$J(n) = \begin{cases} 1 & n = 1 \\ J(n-1) + n^2 - 2n + 1 & n > 1 \end{cases}$$

$$L(n) = \begin{cases} 1 & n = 1 \\ 2L\left(\frac{n}{2}\right) + n & n > 1 \end{cases}$$ (Use integer division; compute the values of *L* for *n* up to 32.)

$$Q(n) = \begin{cases} 0 & n = 0 \\ Q(n-1) + n^3 & n > 0 \end{cases}$$

$$T(n) = \begin{cases} 1 & n = 1 \\ 2T(n-1) + 1 & n > 1 \end{cases}$$

As you complete these Haskell functions, show them to your TA or lab instructor.