CMPSC 497B/CSE 597B HW5. DUE 11/20

Instructions:

You cannot share code with other students or view code written by other students. Do not post code online. Violation = Academic integrity violation.

Your submission will be a zip file containing:

- Your scala code in hw5.scala (in application form, not script form)

Problem 1. *The file calculator.scala contains code for a parser that can parse strings of mathematical expressions, such as "*`log(1 + 2^3.4*9) + 3/exp(45)`*". When we parse such strings using Calculator*`log(1 + 2^3.4*9) + 3/exp(45))` *, it returns a parse tree whose elements are of the base type ASTElement.*

The leaves of these tree are instances of the case class Value (e.g., Value(3.4)). The internal nodes can be the case classes Log, Exp, Power, Add, Subtract, Multiply, and Divide. For example "`log(1 + 2^3)`*" will get parsed as Log(Add(Value(1), Power(Value(2),Value(3)))).*

- *In the file hw5.scala, change the package name appropriately.*
- *Add the appropriate import statements*
- *Fill in the body of the function hw5.evaluate so that it uses the Calculator object (defined in calculator.scala) to parse the arithmetic string s. The result is the root of a parse tree (its type is ASTElement). Evaluate should process this parse tree to obtain the final result of the numerical expression. Use case pattern matching.*

Problem 2. *Write code for the following questions using the functional paradigm.*

- *Fill in the function hw5.pairsToAdjList which takes as input the name of a file of directed graph edges with format:*

  ```
  1, 2
  1, 3
  3, 4
  ```

  and outputs a file of the adjacency list with format:

  ```
  1: 2, 3
  3: 4
  ```

- *Fill in the function hw5.adjListToPairs which does the reverse: converts adjacency lists to directed edges.*
- *Important: for both parts of this question, you cannot use any for/while loops and no mutable state (so you will first need to turn the iterator over input lines into a List). You can only use the various methods that scala collections can have (e.g, map). This restriction includes turning the adjacency list into a string and reading/writing files.*