CS 375
Spring 2021

Homework 3: GP Representation

~~Due 11:59 PM, Monday, March 29~~

We will write this assignment together in class, to give you an easy week as a "virtual spring break" in the middle of this crazy semester. Before class on Monday, March 22, you should read carefully through the assignment and start thinking about how we might program it in Python. The rest of the assignment is written as if you were actually doing it, so I can use it in future years. ***To be clear, there is nothing you need to do for this assignment outside of class after reading through the assignment.***

In this assignment you will create the classes and functions necessary to generate tree-based programs for use in genetic programming. These will include methods for initializing and executing such programs. These programs will then be used in Project 3 to implement a fully-function genetic programming system. I have provided some basic code to create the function set and a list of variables in `hw3.py`. You will add to this file.

Your code should:
- Have your names at the top of your program.
- Exhibit good programming style.
- Have appropriate documentation (i.e. docstrings and comments).
- Have a file docstring that describes how to use your program.

~~Your work on this project includes your code and paper. You can copy the starter code for this assignment to your GitHub at: . You will submit your code by pushing it to GitHub. Please submit your individual papers on Gradescope.~~

Coding

1. First, you will create classes for terminal nodes and function nodes in a program tree. Each of these classes needs to have the following methods:
   a. `__init__` - Terminal nodes should take their value as a parameter, and function nodes should take a function symbol and a list of its child nodes as parameters.
   b. `__str__` - Make a Lisp-like representation of the tree.
   c. `eval` - Given a dictionary of variable assignments, execute the program and return a value.
   d. `tree_depth` - Returns the total depth of this tree rooted at this node.
   e. `size_of_subtree` - Gives the size of the subtree of this node, in number of nodes.

2. Write a function `random_terminal` that half of the time returns a random variable and the other half of the time returns a random float in the range [-10.0, 10.0].

3. Write a function `generate_random_program` that uses Ramped Half-and-Half to

generate random programs. You will likely want to write functions for Full and Grow tree generation methods

~~Paper~~

~~Experiment with your individuals. What types of individuals can you generate randomly?~~