CS2006 Python Project 1 -
Classes and Iterators in Python

Matriculation Numbers: 160001362, 160014817, 160013384 (Group 14)

06/03/2018

Contents

1 Summary of Functionality

This practical specified the development of a mathematical system to explore discrete mathematical structures using the programming language Python.

The provided README file gives detailed instructions on what the solution can do and how to run each command.

The following functionality has been implemented:

1.1 Basic Specification:

All requirements from the basic specification have been implemented. They are as follows:

- An implementation of the twisted integers data structure which supports addition and multiplication given by the following rules:
  - a $\bigoplus$ b = (a + b) mod n
  - a $\bigotimes$ b = (a + b + a b) mod n
- Exceptions that are used to check that user input is valid.
- Unit tests to ensure the correctness of the solution.

1.2 Additional Requirements:

From the suggested additional requirements, all of the easy, medium and hard requirements have been implemented:

1.2.1 Easy

- Easy Requirement 1 - The function 'mulEqualToOne(n)' has been developed in the checker.py file which calculates for a given n all elements x ϵ Zn such that x $\bigotimes$ x = 1, where 1 ϵ Zn
- Easy Requirement 2 - For a given n functions (in checker.py) have been developed to check whether the following properties hold for all x, y, z ϵ Zn (each of which returns a boolean signifying whether the properties hold):
  - x $\bigoplus$ y = y $\bigoplus$ x - 'isCommutativeAdd(n)'
  - x $\bigotimes$ y = y $\bigotimes$ x - 'isCommutativeMul(n)'
  - (x $\bigoplus$ y) $\bigoplus$ z = x $\bigoplus$ (y $\bigoplus$ z) - 'isAssociativeAdd(n)'
  - (x $\bigotimes$ y) $\bigotimes$ z = x $\bigotimes$ (y $\bigotimes$ z) - 'isAssociativeMul(n)'
  - (x $\bigoplus$ y) $\bigotimes$ z = (x $\bigotimes$ y) $\bigoplus$ (y $\bigotimes$ z) - 'isDistributive(n)'

- Easy Requirement 3 - The file twisted_integers.py contains a class TwistedIntegers which implements a data structure representing Zn with respect to the operations a $\bigoplus$ b = (a $\bigoplus$ b) mod n and a $\bigotimes$ b = (a $\bigotimes$ b) mod n and contains the methods __init__, __str__ and size where size returns the number of elements in Zn.

1.2.2 Medium

1.2.3 Hard

1.3 Further Features:

2 Design and Implementation

3 Evidence of Testing

4 Known Problems

5 Problems Overcome

6 Summary of Provenance

6.1 Code Implemented by the Group

6.2 Code Modified From the Provided Source Files

6.3 Code Sourced From Elsewhere

7 Conclusion