Unitec Institute of Technology
TE WHARE WĀNANGA O WAIRAKA

School of Computing, Electrical and Applied Technology

ISCG6426
Data Structures and Algorithms
Lab 2
Semester 2, 2020

Due Date: 8:00PM Fri 09/10/2020

Total Marks: 100
Course Weighting: 10%

Lab 2

Aim
You are required to implement the following data structures.
Skeleton code for part 1 is provided at https://github.com/kris-classes/6426

Instructions

Part 1 – Test-driven Development of Data Structures

Implement the Custom Exceptions (13 marks)
Implement the Recursion base-case (2 marks)
Implement the Stack data structure (19 marks)
Implement the Queue data structure (19 mark)
Implement the Tuple data structure (12 marks)
Implement the Dictionary data structure (18 marks)
Implement the Set data structure (40 marks)
Implement the NaivePriorityQueue data structure (29 marks)

Part 2 – Creation and Testing of Data Structures (using pytest and pyexpect)

Implement a list-based Heap without using the **heapq** module (just use a list).
Implement tests that verify its correctness.
Implement a heap-based Priority Queue using your code from ex8.
Implement tests for the heap-based priority queue to verify its correctness.
Implement an AVL tree.
Implement tests that verify its correctness.

Delivery

- Add your solutions to a 'Labs/Lab2' directory in your 6426 GitHub repository.
- Please **use a separate commit** for each test you solve in the assignment.
- I will clone everyone's repository at 8PM on the due date, then again at 24, 48, and 72 hours after the due date.

Marking Schedule

Task	Marking Criteria	Marks	Given	Comments
Exceptions	Correct TupleException Correct TupleIndexException Correct TupleModifyException Correct StackException Correct StackIsEmptyException Correct StackIsFullException Correct QueueException Correct QueueIsEmptyException Correct QueueIsFullException Correct DictionaryException Correct DictionaryKeyNotFoundException Correct SetException Correct SetElementNotFoundException	3		
Recursion	Correct Recursion base-case	2		
Stack	Correct __str__ method Correct __repr__ method Correct size method Correct is_empty method Correct is_full method Correct push method Correct peek method Correct pop method	5		
Queue	Correct __str__ method Correct __repr__ method Correct size method Correct is_empty method Correct is_full method Correct put method Correct peek method Correct get method	5		
Tuple	Correct __str__ method Correct __repr__ method Correct __getitem__ method Correct __setitem__ method Correct __add__ method Correct count method Correct index method	5		
Dictionary	Correct __str__ method Correct __repr__ method Correct __getitem__ method Correct __setitem__ method Correct get method Correct keys method Correct values method	10		

	Correct items method			
Set	Correct __init__ method Correct __str__ method Correct __repr__ method Correct __contains__ method Correct add method Correct remove method Correct isdisjoint method Correct union method Correct intersection method Correct difference method Correct symmetric_difference method	10		
NaivePriorityQueue	Correct __str__ method Correct __repr__ method Correct size method Correct is_empty method Correct is_full method Correct put method Correct front method Correct peek method Correct get method	10		
Heap	Correct heap using a list Correct tests for heap operations	15		
Heap_PriorityQueue	Correct priority queue using ex8 heap Correct tests for HeapPQ operations	15		
AVL Tree	Correct AVL tree Correct tests for AVL tree operations	20		
	100			

Late Submission of Assignments:

Assignments submitted after the due date and time without having received an extension through Affected Performance Consideration (APC) will be penalised according to the following:

- 10% of marks deducted if submitted within 24hrs of the deadline,
- 20% of marks deducted if submitted after 24hrs and up to 48hrs of the deadline,
- 30% of marks deducted if submitted after 48hrs and up to 72hrs of the deadline,
- No grade will be given for an assignment that is submitted more than 72hrs after the deadline.

Assistance to other Students:

Students themselves can be an excellent resource to assist the learning of fellow students, but there are issues that arise in assessments that relate to the type and amount of assistance given by students to other students. It is important to recognise what types of assistance are beneficial to another's learning and also what types of assistance are unacceptable in an assessment.

Beneficial Assistance:

- Study Groups
- Discussion
- Sharing Reading Material
- Reading the available online and library resources

Unacceptable Assistance:

- Working together on one copy of the assessment and submitting it as own work
- Giving another student your work
- Copying someone else's work, this includes work done by someone not on the course
- Changing or correcting another student's work
- Copying from books, the Internet etc. and submitting it as own work; anything taken directly from another source must be acknowledged correctly; show the source alongside the quotation