Final Report

Introduction

This is my final report about the testing I had done on the library of data structures using TSTL . The library was cloned from https://github.com/nryoung/algorithms .
I have tested binary search tree, stack , queue and linked list structures.

Binary Search Tree is a binary tree where each node has a key and the value of each key should be larger than the keys on the left side of the node and should be smaller than the keys on the right side of the node.
In the module I have chosen, the binary search tree has the following functions –
1. Size - The size function returns the number of nodes in binary search tree
2. Is_empty – This function returns true if the bst is empty or else returns false
3. Get – Returns the value that is paired with the key
4. Contains – Returns true if the bst contains a key , else returns false
5. Put – It adds a new key value pair
6. Min_key – It returns the minimum value key in the bst
7. Max_key – It returns the maximum value key in the bst
8. Floor_key – It returns the biggest key that is less than or equal to the given key value
9. Ceiling_key – It returns the node with the smallest key that is greater than or equal to the given key
10. Select_key – Returns the key with rank equal to the rank that is passed as a parameter
11. Rank – It returns the number of keys less than the given key
12. Delete – Removes the node with the key equal to the key passed as a parameter
13. Delete_min – Removes the key value pair with the smallest key in the tree
14. Delete_max – removes the key value pair with the largest key in the tree
15. Keys – This function returns all the keys in ascending order

Testing

I have written three test cases for this module.
I have tested put , contains, is_empty , delete, delete_max functions of the binary search tree.
In the first test, I put a key value pair in the tree , and check if the inserted key contains in the tree and prove that the is_empty is false.
In the second test, I check if the tree is empty , I put a key value pair and delete the key. Then check if tree empty is true.
In the third test, put a key value pair, check contains is true , delete a key and check contains that key is false.
In the fourth test , I put a key value pair , store the min key in a variable and then store min node in a variable . Then check if the min key is equal to min node of key.
Check min node of key is min key and max node of key is max key.
Store min key in a variable and store the delete min in a variable and compare both the variables.

Code Coverage
43.7751004016 PERCENT COVERED
104 BRANCHES COVERED
72 STATEMENTS COVERED

Coverage .out

```
Name                Stmts   Miss Branch BrPart  Cover   Missing
----------------------------------------------------------------
binary_search_tree.py     169     97     80      7    44%   14-19, 27-32, 35, 41-61, 63, 72, 82, 92, 111, 121, 128, 135, 145, 149, 156, 163, 173, 177-216, 231-304, 306, 325, 335, 343, 354-395, 62->63, 127->128, 144->145, 155->156, 172->173, 224->231, 305->306
```

The second code , I have tested in the queue.
Queue has 4 functions
1. Add – Adds element as the last element of the queue
2. Remove – Removes element from the front of the queue and return the value
3. Is_empty – Returns a Boolean value that indicates if the queue is empty
4. Size – returns size of the queue

I have just tested the add , is_empty and remove functions.
In one test case , I added an element to queue.
And checked if the queue is not empty , remove the element that was added.
In a new test case , I added an element and compared current queue size with the previous size incremented by 1. Also , removed an element from queue and checked the size with the previous size decremented by 1.

Code Coverage for queue test case
23.0769230769 PERCENT COVERED
6 BRANCHES COVERED
3 STATEMENTS COVERED

Coverage.out

```
Name       Stmts   Miss Branch BrPart  Cover   Missing
-------------------------------------------------------
queue.py      13     10      0      0    23%   13-19, 22, 29, 37-49
```

Stack
The third library that I have tested is stack
Stack has the following function –
1. Add – Adds element at last
2. Remove – Removes element from the last return value
3. Is_empty – Returns a Boolean value that indicates if the stack is empty .
1 if stack is empty and 0 when stack is not empty
4. Size – returns size of the stack

I have written test cases to test all the functions in the .py file.
I add a value to the stack. Then check if the stack is not empty , I remove the element.
Check , if stack is not empty and store the stacklist in a variable called length. Get the value at the end of the array in a variable val and compare the value by doing a remove function on the stack .
Another , test case is adding a value and checking the size by incrementing the previous size by

1.

Code Coverage
25.0 PERCENT COVERED
6 BRANCHES COVERED
3 STATEMENTS COVERED

Coverage.out

```
Name       Stmts   Miss Branch BrPart  Cover   Missing
-------------------------------------------------------
stack.py      12      9      0      0    25%   12-15, 18, 25, 33-45
```

Linked List
Linked List has the following functions –
1. Add – Adds element to the linked list
2. Remove – Removes element from the list
3. Search – Search for value in list.
4. Size – returns size of the list

I wrote test cases to test all the functions of linked list available.
If the size of the linked list is not 0 and then I search for an node . If it is present then I remove the node and check the size to previous size – 1.
I then add a node and check the current size with the previous size incremented by 1.

Coverage
50.9090909091 PERCENT COVERED
31 BRANCHES COVERED
22 STATEMENTS COVERED

Coverage.out

```
Name                    Stmts   Miss Branch BrPart  Cover   Missing
-----------------------------------------------------------------
singly_linked_list.py      45     23     10      0    51%   14-16, 20-
26, 29-35, 39, 49, 73-93
```

I extensively tested these modules using tstl, but I couldn’t find any bugs as this library is widely used and well written.

The test cases for each data structure cover almost every function in the script.And also , all the corner cases have been covered.
So , I assume it is a good attempt to test the module using tstl.

TSTL is a very good testing tool to test python libraries. It has all the interfaces required for testing a code. It is also the best way to learn about automated testing. The code coverage gives the idea of how efficient the test case is .

Overall , it was a great learning experience to use tstl.