Student: Weiyu Lin

CS 562 Applied Software Engineering

Tested Python Library: "Algorithms" on Github

Url: https://github.com/nryoung/algorithms

- **Describe some bugs you found (in the SUT or your tester)**

No bug was found so far. This section will be updated if I find any bug in the Python library I test.

- **Explain progress to date -- how has your test system improved from the start?**

Data Structure module and Searching Algorithm module are two testing object I have set on my "Testing Project Proposal".

1. **Data Structure**

This section contains 9 modules which are : binary search tree, digraph.py, queue.py, stack.py, undirected graph.py , union find.py, union find by rank.py, union find with path compression.py. Except digraph.py and undirected graph.py modules, all the modules have been tested. I have tested the functions in each module by TSTL. For example, For stack testing, I compared size() function and recall len(stack.list) function to check the correctness of size() function.

~<stack>.add(~<int>) => <stack,1>.size() == len(<stack,1>.stack_list)

I have checked the statue of stack by is_empty() function after initial it by:

<stack> := stack.Stack()

~<stack>.is_empty() and True

I have inserted integer into stack data structure and recalling series of functions in stack to see if all the functions(stack.add(), stack.remove, stack.isempty(), stack.size(),) works as expected.

As we all known, the essence of data structures is properties. For the next step, I am committed to testing the properties of each data structure. For example, FIFO (first in first out) to stack, FILO (first in last out) to queue.

2. **Searching Algorithm**

This section contains 6 modules, which are: binary search.py, bmh search.py, breadth first search.py, depth first search.py, kmp search.py, rabinkarp search.py. I have just start this section. Few of modules are primary tested (binary search, bmh breadth first search). I merely input some data and output specific one to check the correctness. Since I am not familiar with kmp search and rabinkarp search, I haven't figured out a good way to test these two algorithms.

For the next step, I will complete the testing of searching algorithms mentioned above. I will focus on the correctness in searching algorithm. In the meanwhile, I will try to check if each searching algorithm meets the theoretical time complexity.

- **Estimate your progress by end of term**

Base on the current schedule, most of data structure and searching algorithms can be finished. I will focus on my confusion part (graph, undirected graph, kmp search and rabinkarp seach) to see if there is any progression.

- **Discuss quality of the SUT**

My conclusion is the quality of SUT for data structure section is stable cause I haven't found any bugs on that part. Since I haven't make too much effort on searching algorithm part, I can't give opinions about how the SUT quality of that part.

- **Talk about code coverage**

As the result displayed in the terminal, my code coverage is about 45 percent currently. I believe this result is not good enough. New data will be updated when any progression occurs.

Word count: 504