

CS 562 Winter 2016

Qi Wang

ID: 932-439-151

wangq5@oregonstate.edu

Bug/Progress Report

1. Describe some bugs you found

When testing the double linked list program, I found the author is very thoughtful, and consider many situations and cases. So I did not find any bugs for this program when using the tstl file to test so far. However, I found some not very good situation when running the python files. For example, when I call a. deleteData(a.insert(4)), it returns a false. It should returns a false, but it actually inserts twice of 4 to the doubly linked list. So far, I did not write a tstl code to test this. So I did not find this is a bug in the tstl.

2. Explain progress to date

In the beginning of this project, I have tested the insert() and append() functions of the double linked list python program. The testing function of tstl file looks like the following code:

```
~<DOUBLELINKEDLIST>.insert(<INT>)=>(DLLlength(<DOUBLELINKEDLIST, 1>)==PRE<(DLLlength(<DOUBLELINKEDLIST, 1>))+1)
```

In order to test the insert function, firstly, I declared 10 integers and 3 DOUBLELINKEDLIST to store states of the SUT. Then I initialize the range of the integers and an empty double kinked list, which I named it as DOUBLELINKEDLIST. The DOUBLELINKEDLIST is the target double linked list to do the functions. For the DOUBLELINKEDLIST, there are two pointers for one data which are prev and next. On the other hand, in order to test the insert function precisely, I defined a function which name is DLLlength(). When inserting some INT numbers to the double linked list, the total length of the double linked list should be equal to the final total value of the entries of the double linked list. If insert function is correct, they should be equal.

Then I also test the append function of the double linked list. The code I used to test the append function is as following:

```
~<DOUBLELINKEDLIST>.append(<INT>)=>(DLLlength(<DOUBLELINKEDLIST, 1>)==PRE<(DLLlength(<DOUBLELINKEDLIST, 1>))+1)
```

This is quite similar to the code that tests the insert function. Their only difference is the append function can add new data to the list, but it adds the new data to the tail of the list, but the insert function can only add new data after previous pointer, in other word, it means it would add new data before the old data.

For now, I defined some functions that used to test other main functions of the double linked list python program to test deleteIndex() function. I define a function in the tstl file to return the value of the data in the double linked list. When testing the deleteIndex function, for example, when deleting the data's index is two, first call the function to return the value. Then if deleteIndex function works successfully, the return value should be false.

3. Estimate your progress by the end of term

By the end of this term, I think I will write more functions to test the rest of the functions of the double linked list python program. For example, deleteData() function, deletePtr() function, deleteAllData() function, deleteList() function, copyList() function, insertAfterEveryData() function, insertBeforeEveryData() function.

For the deleteList() function, it is used for deleting the list. so I may check if the list is empty which I defined in the TSTL file.

For the deleteData() function, it is used for deleting specific the data in the list. I may first define a function to get the value of the linked list. When testing the deleteData function, the value of the deleting data should not exist in the linked list, the function will return a false. When it returns a false, it means the deleteData function is successful.

For the insertBeforeEveryData function, I may firstly define a function to return the number of the list. Then when testing the insertBeforeEveryData function, ensure the final list's Dllength is the length of original list plus the number of new data.

For the other functions, I think they might be similar to the above functions.

4. Discuss quality of the SUT

I think the quality of SUT is fine. I think it can easily test the basic data structure for instance the linked list and stack. For this project, since the author of this python program designed very well and very thoughtful, so I did not found any bugs for now. However, I think there should be a bug for any program. So I will keep improving the tstl code to find it.

5. Talk about code coverage, if possible

For the beginning of the `tstl` file, I only test the `insert` and `append` functions. The code coverage is only 8 percent. For now, after testing of `deleteIndex`, it only covers 12 percent. So I think by the end of this term, I will at least design the `tstl` to cover more than 50 percent. I may find some bugs in the future test.