Progress report

For the last project, I tested three programs by using TSTL, singly linked list, double linked list and adjacency list.

1. Bugs

I have not found any bug yet.

2. Progress and estimate

I wrote some codes in order to test linked list. The linked list is a linear collection of data elements, called nodes pointing to the next node by means of pointer. It is a data structure consisting of a group of nodes which together represent a sequence. Under the simplest form, each node is composed of data and a reference to the next node in the sequence; more complex variants add additional links. This structure allows for efficient insertion or removal of elements from any position in the sequence. To test the LinkedList_Single.py, I create 10 different INT numbers and 3 singly linked list. The value of the INT is from 1 to 20. Then I initialized an empty linked list. For this project, I mainly test the insert function. The principal benefit of a linked list over a conventional array is that the list elements can easily be inserted without reallocation or reorganization of the entire structure, and it allows insertion of nodes at any point in the list. Singly linked lists contain nodes which have a data field as well as a 'next' field, which points to the next node in line of nodes. So for the singly linked list, the insert function can only insert the integer after <HEAD>, that means if I insert 1,2,3 in order, the singly linked list I will get is 3->2->1. So I design the test code based on that. In addition, I also create a function that indicate how many integers are already in the singly linked list, so that I can also check whether the singly linked list is full or not when I call the insert function.

After done those I mentioned, I added some extra functions to my TSTL file in order to test other function in singly linked list, like insertAfterIndex(), deleteList(), and so on.

For the doubly linked list, the progress of this part is similar with singly linked list. The only difference between singly linked list and doubly linked list is the doubly linked list is a linked list that each node contains, besides the next-node link, a second link field pointing to the 'previous' node in the sequence. So there will be two pointer: <prev> and <next> for one data.

I also tested the adjacency list. The adjacency list is a collection of unordered lists used to represent a finite graph. Each list describes the set of neighbors of a vertex in the graph. This is one of several commonly used representations of graphs for use in computer programs. For this part, I tested the addVertex function and addEdge function. First, I defined 10 vertices and 3 adjacency lists. Second, I defined the range of the vertices. Finally, I initialized the adjacency list and test the addVertex function and addEdge function. For the addVertex function, it is simply add a new vertex in the vertexList, return False if the vertex already exists. For the addEdge, it is used for add vertices from one position to the target position.

I think there are many things I can do for my project. I only test the insert function for those three programs. There are lots of functions I can test, for example, in linked list, there are insertBeforeIndex, insertAfterIndex, deleteIndex, deleteList and so on. Likewise, there are also many functions in the adjacency list.

3. Discuss quality of the SUT

I think System under test (SUT) is OK, I still have not found all the bugs.

4. Coverage

For now, my coverage is only about 9 percent. I will improve in future.