

CSC111 Winter 2023 Assignment 1: Linked Lists and Blockchain

Vrinda Subhash

January 30, 2023

Part 1: Linked List Debugging Exercise

1. (a) The first error I found was that the code does not properly swap the first and last node in the parallel assignment. In the code, `self.first` gets assigned to `curr`. At that point, `curr` is the last node and points to `None`. So, the `LinkedList` will just be the `curr` node, and nothing else. All of the other nodes are gone, and then it is just the last node. To fix it, instead of switching the nodes, you can just update the node items. You can do this by making a temp variable to hold the value of the first node's item, and then making the first node's item the value of `curr.item`, and then setting `curr.item` to be temp. This will properly swap the first and last items.
(b) Complete your work for part (b) in `a1_part1.py`.
2. (a) The second error I found was that if this function was called on an empty linked list, an `AttributeError` is raised. In the function docstring it says that nothing should be done if there are fewer than two items in the linked list. But, the function does not have any code to deal with those special cases, and so when called on an empty linked list, the while loop will check `curr.next`, but `curr` is of type `None`, and `None` does not have an attribute "next", so an `AttributeError` is raised. To fix this error, there needs to be an if-statement at the beginning of the function to check for the case of an empty linked list and linked list with one node. In those cases, nothing should be done, so if you add a 'return' with no value in those if-statements, the function will just stop and be done as it should.
(b) Complete your work for part (b) in `a1_part1.py`.
3. (a) Complete your work for part (a) in `a1_part1.py`.
(b) The third test case, which is a linked list with 1 node, passes despite the function having errors. The first error that I identified is that the first and last nodes do not get correctly swapped; the linked list once mutated will just end up being the last node in the linked list. But, in a linked list with just one node, that one node will be the last node and so then the linked list once mutated will still be that same node and so the function will have appeared to do what it was supposed to for a linked list with fewer than 2 items, which is nothing. The second error I identified is that there is not a if-condition to handle the case of there being less than 2 nodes in the linked list. In those cases, nothing should be done to the linked list. Since this test case just had one item in the linked list, `curr.next` did exist and was `None`, so then no `AttributeError` was raised and then the function went on to the swap, which as I explained above works on linked lists with just one node. So despite the swap not properly working and there not being a special condition to handle linked lists with less than 2 nodes, this test case (a linked list with just 1 node) will still pass.
4. Complete your work for this question in `a1_part1.py`.

Part 2: Blockchain and Cryptocurrencies

Complete this part in the provided `a1_part2.py` starter file. Do **not** include your solutions in this file.