Solution Review: Find the Length of a Linked List

This review provides a detailed analysis of the solution to the Find the Length of a Linked List challenge.

We'll cover the following

Solution: Linear Iteration

main.py

LinkedList.py

Node.py

```
from Node import Node
from LinkedList import LinkedList


def length(lst):
    # start from the first element
    curr = lst.get_head()
    length = 0

    # Traverse the list and count the number of nodes
    while curr:
        length += 1
        curr = curr.next_element
```

```
14      return length
15
16
17  lst = LinkedList()
18  lst.insert_at_head(4)
19  lst.insert_at_head(3)
20  lst.insert_at_head(2)
21  lst.insert_at_head(1)
22  lst.insert_at_head(0)
23  print(length(lst))
24
```

The logic is very similar to that of the `search` function. The trick is to iterate through the list and keep count of how many nodes you've visited. This count can be kept in the `length` variable.

Time Complexity

Since this is a linear algorithm, the time complexity will be *O(n)*.