



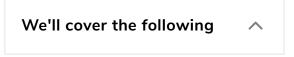






## Solution Review: Find the Length of a Linked List

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



- Solution: Linear Iteration
  - Time Complexity

## Solution: Linear Iteration #

```
main.py
LinkedList.py
Node.py
    from Node import Node
     from LinkedList import LinkedList
  3
  4
  5
     def length(lst):
         # start from the first element
  7
         curr = lst.get_head()
  8
         length = 0
  9
         # Traverse the list and count the number of nodes
 10
 11
         while curr:
 12
              length += 1
 13
              curr = curr.next_element
```

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).

Interviewing soon? We've partnered with Hired so that

X companies apply to you instead of you applying to them. See how ① ← Back Next  $\rightarrow$ Challenge 4: Find the Length of a Link... Challenge 5: Reverse a Linked List ✓ Comple C



