

# Solution Review: Reverse a Linked List

This review provides a detailed analysis of the different ways to solve the Reverse a Linked List challenge.

## We'll cover the following



- Solution: Iterative Pointer Manipulation
  - Time Complexity

## Solution: Iterative Pointer Manipulation #

main.py

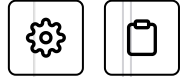
LinkedList.py

Node.py

```
1 from LinkedList import LinkedList
2 from Node import Node
3 def reverse(lst):
4     # To reverse linked, we need to keep track of three things
5     previous = None # Maintain track of the previous node
6     current = lst.get_head() # The current node
7     next = None # The next node in the list
8
9     #Reversal
10    while current:
11        next = current.next_element
12        current.next_element = previous
13        previous = current
```



```
14     current = next
15
16     #Set the last element as the new head node
17     lst.head_node = previous
18     return lst
19
20 lst = LinkedList()
21 lst.insert_at_head(6)
22 lst.insert_at_head(4)
23 lst.insert_at_head(9)
24 lst.insert_at_head(10)
25 lst.print_list()
26
27 reverse(lst)
28 lst.print_list()
```



The brain of this solution lies in the loop which iterates through the list. For any `current` node, its link with the `previous` node is reversed and `next` stores the next node in the list:

- Store the `current` node's `next_element` in `next`
- Set `current` node's `next_element` to `previous` (reversal)
- Make the `current` node the new `previous` so that it can be used for the next iteration
- Use `next` to move on to the next node

In the end, we simply point the `head` to the last node in our loop.

## Time Complexity #

The algorithm runs in  $O(n)$  since the list is traversed once.



Hopefully, you've got a good idea of pointer manipulation by now. The next challenge will be a little trickier, so don't be afraid to test yourself.

[← Back](#)[Next →](#)[Challenge 5: Reverse a Linked List](#)[Challenge 6: Detect Loop in a Linked L...](#)[Mark as Completed](#)[Report an Issue](#)[Ask a Question](#)

([https://discuss.educative.io/tag/solution-review-reverse-a-linked-list\\_\\_introduction-to-linked-lists\\_\\_data-structures-for-coding-interviews-in-python](https://discuss.educative.io/tag/solution-review-reverse-a-linked-list__introduction-to-linked-lists__data-structures-for-coding-interviews-in-python))