



Solution Review: Remove Duplicates from a Linked List

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

We'll cover the following

- Solution:
- Time Complexity

Solution:

main.py

LinkedList.py

Node.py

```
1 from LinkedList import LinkedList
2 from Node import Node
3
4
5 def remove_duplicates(lst):
6     if lst.is_empty():
7         return None
8
9     # If list only has one node, leave it unchanged
10    if lst.get_head().next_element is None:
11        return lst
12
13    outer_node = lst.get_head()
```

```

14     while outer_node:
15         inner_node = outer_node # Iterator for the inner loop
16         while inner_node:
17             if inner_node.next_element:
18                 if outer_node.data == inner_node.next_element.data:
19                     # Duplicate found, so now removing it
20                     new_next_element = inner_node.next_element.next_element
21                     inner_node.next_element = new_next_element
22             else:
23                 # Otherwise simply iterate ahead
24                 inner_node = inner_node.next_element
25         else:
26             # Otherwise simply iterate ahead
27             inner_node = inner_node.next_element
28         outer_node = outer_node.next_element

```

In this implementation, we check each node against the remaining list to see if a node contains an identical value.

outerNode iterates through the outer loop, while **innerNode** checks for duplicates. Whenever a duplicate is found, it is removed from the list

```
inner_node.next_element = inner_node.next_element.next_element
```

This algorithm is the most basic way to remove duplicates from a linked list. Hence, it is not very efficient.

A significantly smarter solution is explained below.

Time Complexity

The nested while loops increase this program's complexity to $O(n^2)$.

Note: The solution provided above is not the optimal solution for this problem. We can write a more efficient solution using hashing. We will

cover that approach in Hashing Chapter: [Challenge 11](#)



Interviewing soon? We've partnered with Hired so that companies apply to you instead of you applying to them. [See how](#) ⓘ



← Back

Next →

Challenge 8: Remove Duplicates from ...

Challenge 9: Union & Intersection of Li...

✓ Completed



Report an Issue

