Challenge 11: Remove Duplicates from Linked List

In this lesson, you must figure out the Pythonic solution for removing duplicates from a linked list.

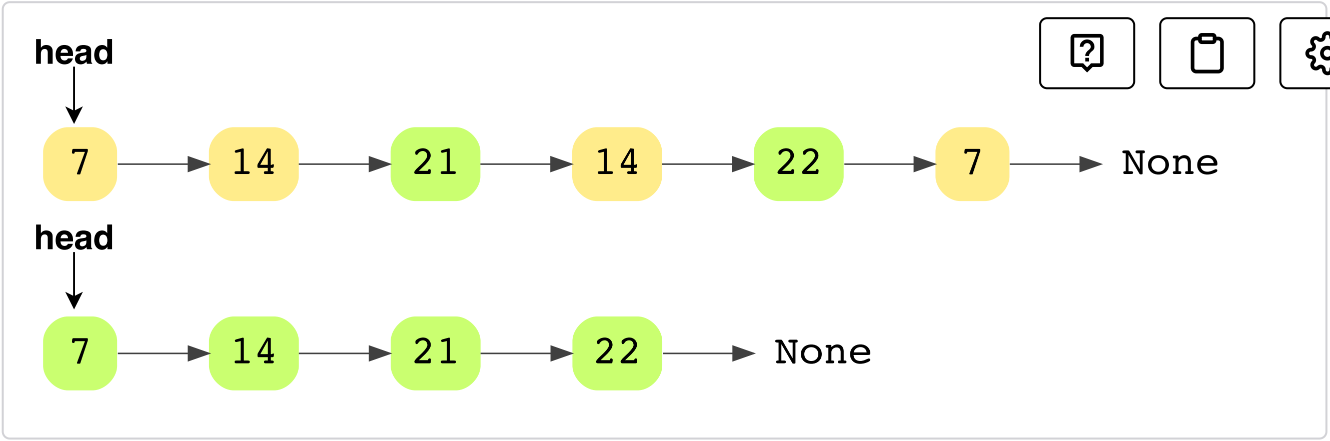
We'll cover the following

Problem Statement

You will now be implementing the `remove_duplicates()` function. When a linked list is passed to this function, it removes any node which is a duplicate of another existing node.

You have already seen this challenge previously in chapter 3 of this course. Here you would use HashTables for a more efficient solution.

You can see an example below:



Input

A linked list.

Output

A list with all the duplicates removed.

Sample Input

```
LinkedList = 1->2->2->2->3->4->4->5->6
```

Sample Output

```
LinkedList = 1->2->3->4->5->6
```

Coding Exercise

Once again, there are several ways to solve this problem. As you play around with this algorithm, you'll learn that some approaches are much more efficient compared to others.

We'll take a look at some of the solutions, so don't worry if you get stuck.

Good luck!

main.py

LinkedList.py

Node.py

```
from LinkedList import LinkedList
from Node import Node
# Access head_node => list.get_head()
# Check if list is empty => list.is_empty()
# Delete at head => list.delete_at_head()
# Delete by value => list.delete(value)
# Search for element => list.search()
# Length of the list => list.length()
# Node class  { int data ; Node next_element;}


def remove_duplicates(lst):
    # Write - Your - Code
    return
```

Back
Solution Review: Detect Loop in a Link...

Next
Solution Review: Remove Duplicates f...

✓ Mark as Completed

Report an Issue