

Exercise: Remove Duplicates

Challenge yourself with an exercise in which you'll have to remove duplicates from a doubly linked list.

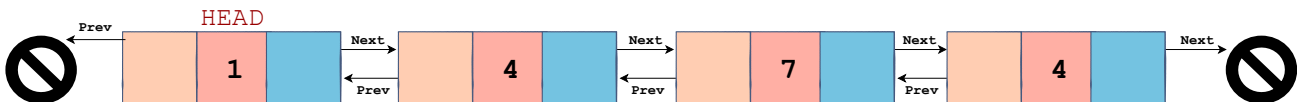
WE'LL COVER THE FOLLOWING ^

- Problem
- Coding Time!

Problem

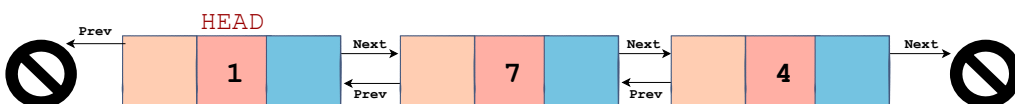
In this exercise, you are required to remove the duplicates from a doubly linked list.

Doubly Linked List: Remove Duplicates



1 of 2

Doubly Linked List: Remove Duplicates





In the exercise widget, the `delete_node` method has been given to you. It is a slight modification of the `delete` method we covered in one of the previous lessons. You have to figure out the modification yourself. The hint is to recall the `remove_duplicates` lesson from one of the previous chapters.

Coding Time!

In the code below, the `remove_duplicates` is a class method of the `DoublyLinkedList` class. You cannot see the rest of the code as it is hidden. As `remove_duplicates` is a class method, please make sure that you don't change the indentation of the code provided to you. You are required to write your solution under the method prototype.

For this exercise, you are not required to return anything. Just remove the duplicates from the linked list using the `delete_node` method.

Good luck!

```
def remove_duplicates(self):
    pass

def delete_node(self, node):
    cur = self.head
    while cur:
        if cur == node and cur == self.head:
            # Case 1:
            if not cur.next:
                cur = None
                self.head = None
                return

            # Case 2:
            else:
                nxt = cur.next
                cur.next = None
                nxt.prev = None
                cur = None
                self.head = nxt
                return

        elif cur == node:
            # Case 3:
```



```
    if cur.next:
        nxt = cur.next
        prev = cur.prev

        prev.next = nxt
        nxt.prev = prev
        cur.next = None
        cur.prev = None
        cur = None
        return

# Case 4:
else:
    prev = cur.prev
    prev.next = None
    cur.prev = None
    cur = None
    return
cur = cur.next
```

