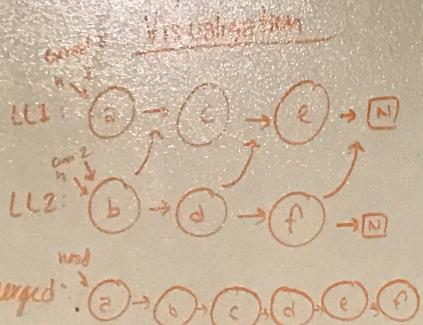


Problem Domain

Gives 2 linked lists, zip them together alternating nodes
 where \rightarrow reference to the next.
 Inputs: Linked lists
 Output: head node

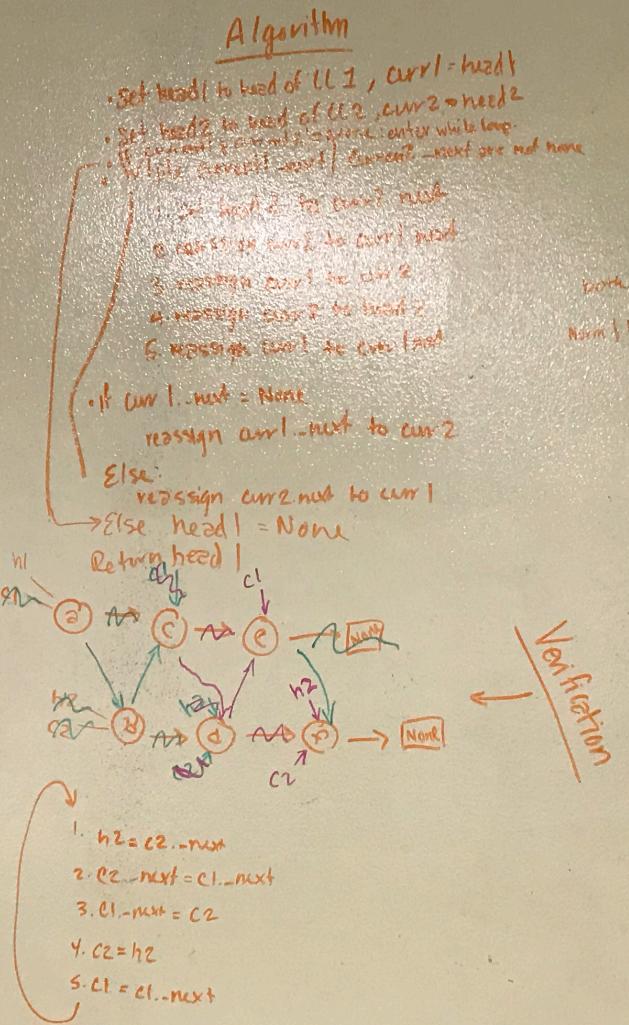


Edge Case

- one list longer than the other
- empty linked list: $\text{head} \rightarrow \text{head}$ if not empty, $\text{head} \rightarrow \text{None}$
- both linked list empty: $\text{head} \rightarrow \text{None}$

Big O

time: $O(n)$
 space: $O(n)$



Code

```

def ll_merge(LL1, LL2):
    head1 = LL1.head
    curr1 = head1
    head2 = LL2.head
    curr2 = head2

    while curr1 & curr2:
        if not curr1 & curr2:
            break
        if not curr1 & not curr2:
            break
        if curr1 & curr2:
            curr1.next = curr2
            curr2 = curr2.next
            curr1 = curr1.next
    if curr1:
        curr1.next = curr2
    else:
        head1 = None
    return head1
  
```