Problem Domain

Zip two linked lists

Example

Input

List 1: 1 → 3 → 5 → 7
List 2: 2 → 4 → 6 → 8

Output

List 1: 1 →2 → 3 → 4 → 5 → 6 → 7 → 8

Algorithm

1. If either list1 or list2 is empty, return the other list.
2. Create a new empty linked list, zipped_list, and a dummy node, tail, pointing to the head of the new list.
3. While both list1 and list2 are not empty, repeat steps 4-5.
4. Append the current node of list1 to the tail of the new list.
5. Append the current node of list2 to the tail of the new list.
6. If there are any remaining nodes in list1 or list2, append them to the tail of the new list.
7. Return the head of the new list, which is the next node after the dummy node.

```python
def zip_lists(list1, list2):
    if not list1:
        return list2
    if not list2:
        return list1

    # create a dummy node to start the new list
    dummy = Node(None)
    tail = dummy

    while list1 and list2:
        # add the current nodes from both lists to the new list
        tail.next = list1
        list1 = list1.next
        tail = tail.next

        tail.next = list2
        list2 = list2.next
        tail = tail.next

    # add any remaining nodes from the input lists to the new list
    if list1:
        tail.next = list1
    if list2:
        tail.next = list2

    # return the new list
    return dummy.next
```

Time & Space Complexity

O(n) Time Complexity:

- The time complexity of this algorithm is O(n), where n is the length of the longer input list.

O(1) Space Complexity:

- The space complexity of this algorithm is O(1), as we only create a constant number of additional nodes to hold the zipped list.