Walkthrough of Code:
The data structure given as an input is a Linked List. To find the union and intersection of the elements would require an easy lookup of each value in either one of the linked lists in the other. A simple and straightforward implementation would involve a dictionary to determine if a node either one of the linked lists is present in the other linked list given the value as an index for the dictionary.

Time Complexity: O(n)
The implementation of union and intersection involves iterating through either of the linked lists first and to map it out on a dictionary. Through iterating through the other linked list, the distinction between the two methods is either to derive the elements in either linked list or append nodes that aren't. Overall, either method involves iteration through the linked lists, or a linear time complexity.

Space Complexity: O(n)
The linked list space complexity varies linearly with the user's input, or the two linked lists referenced.