The ‘len’ method operates in constant time, it simply returns the size which is updated when one inserts, appends, or removes an element to the linked list. The ‘append element’ method also works in constant time. ‘Append element’ is constant time because the linked list is doubly linked, meaning it does not have to ‘current walk’ to the node before trailer, it instead must take one ‘step’ back from trailer and it is always in the right position. The ‘Iter’ method also operates in constant time, all it does is initialize an attribute. The ‘next’ method grabs a value and then updates the variable created in ‘iter’, it is also a constant time method. All of these methods operate in constant time because they take the same number of steps no matter the size of the lips.

In my project I created a ‘current walk’ method to iterate through the linked list. Therefore, any method that calls ‘current walk’ will operate in linear time as the number of steps ‘current walk’ takes increases as the size of the list increases. The methods that use ‘current walk’ include ‘get element at’, ‘remove element at’, and ‘insert element at’. These methods operate in linear time because after they call ‘current walk’ the number of steps they take are constant. The ‘rotate left’ method contains a for loop that is bounded by the size of the list. This means the number of times the for loop iterates increases as the size of the list increases. The rest of the steps in the ‘rotate left’ method are constant, so ‘rotate left’ is a linear time method. The ‘str’ method contains a while loop that will iterate until it reaches the last non-sentinel value in the linked list. The last non-sentinel value is farther away from the beginning of the list the larger the list is, meaning it takes more steps the larger the list is. The rest of the steps in ‘str’ are constant so ‘str’ operates in linear time. The final method is ‘reversed’. The ‘reversed’ method contains two for loops, however neither for loop is nested, so it does not operate in quadratic time. Another note is that the ‘append element’ method operates in constant time if we did not doubly link the list ‘append element’ would have to be linear and therefore ‘reversed’ would be quadratic as the ‘append element’ method within a for loop for this method. All of these methods are linear because they have a loop bound in some way by the size of the linked list, however there are no nested loops, so we have no methods that operate in quadratic time.