



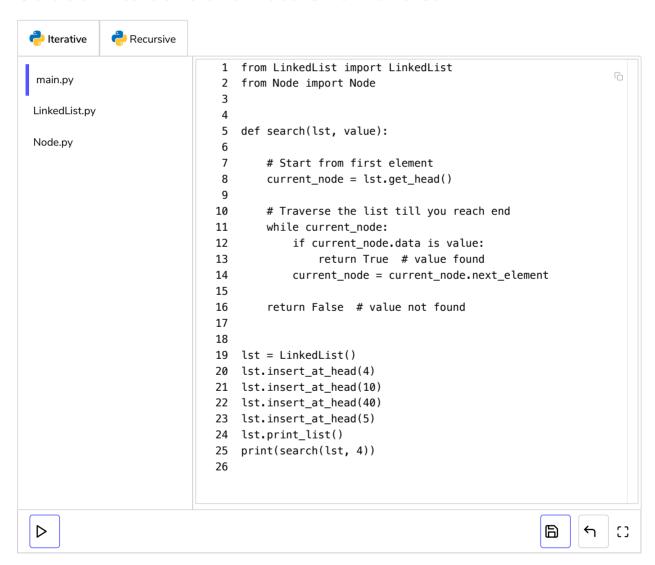
## Solution Review: Search in a Singly Linked List

This review provides a detailed analysis of the different ways to solve the Search in a Singly Linked List challenge.



- Solution: Iterative and Recursive Traversal
  - Time Complexity

## Solution: Iterative and Recursive Traversal #



In both approaches, we traverse through the list, checking whether the current node's data matches our value. The two statements below are equivalent:

```
current_node = current_node.next_element #iterative step
search(node.next_element, value) #recursive step
```

Note that the recursive function takes a node as parameter whereas the iterative version takes the entire list as a parameter.

## Time Complexity #

The time complexity for this algorithm is O(n). However, the space complexity for the recursive approach is also O(n), whereas the iterative solution can do it in O(1) space complexity.

And there you have it. We're done with the **search** operation.

In the next lesson, we will look at how **deletion** works in a singly linked list.

