**Lab 5 PART A:  
Implement a Singly Linked List in Python**

In this assignment, you'll be implementing a singly linked list in Python. Your linked list should have the following methods:

1. \_\_init\_\_(self) - Initializes an empty linked list.
2. is\_empty(self) -> bool - Returns True if the linked list is empty, False otherwise.
3. prepend(self, data: Any) - Inserts a new node containing data at the beginning of the linked list.
4. append(self, data: Any) - Inserts a new node containing data at the end of the linked list.
5. insert\_after(self, target: Node, data: Any) - Inserts a new node containing data after the node target.
6. delete(self, target: Node) -> bool - Deletes the node target from the linked list. Returns True if the deletion was successful, False otherwise.
7. search(self, data: Any) -> Optional[Node] - Searches the linked list for a node containing data. Returns the node if it's found, None otherwise.
8. size(self) -> int - Returns the number of nodes in the linked list.
9. to\_list(self) -> List[Any] - Returns a list containing the data in the linked list in order.
10. print(self) prints all the elements in the linked lists

You'll also need to implement a Node class with the following attributes:

* data - The data stored in the node.
* next - A reference to the next node in the linked list.

Please refer to the attached Skeleton code.

**PART B: big-O analysis  
Perform a big-O analysis of the functions above based on the size of the linked list (T(n) and O(n), n being the size of the array)**