

ADDING 1 TO A LINKED LIST

★ In this problem, we are given a linked list which represents a number. Our job is to add 1 to the number and return the linked list which contains our result.

Bruteforce solution is to reverse the linked list, add 1 by maintaining a carry and after adjusting all required nodes, we reverse it again. Time Complexity is $O(3N)$ and space complexity is constant.

Optimal Solution involves the use of recursion (because it has backtrack-ing we do not need to reverse). We just maintain a carry and keep on adding.

C++ :

```
Node * add1ToLinkedList(Node * head) {  
    int carry = adder(head)  
    if (carry == 1) {  
        Node * newNode = new Node(1);  
        newNode->next = head;  
        return newNode;  
    }  
    return head;  
}
```

```
Node * adder(Node * head) {  
    if (head == nullptr) {  
        return nullptr ;  
    }  
    int carry = adder(head->next) ;  
    head->data = head->data + carry ;  
    if (head->data < 10) {  
        return 0 ;  
    }  
    head->data = 0 ;  
    return 1 ;  
}
```