Challenge 10: Return the Nth node from End

Returning the Nth node from the start of a linked list is easy. Can you return Nth node from the end of a list?

We'll cover the following

Problem Statement:

In the `find_nth` function, a certain **N** is specified as an argument. You simply need to return the node which is **N** spaces away from the `None` end of the linked list.

Input

A linked list and a position *N*.

Output

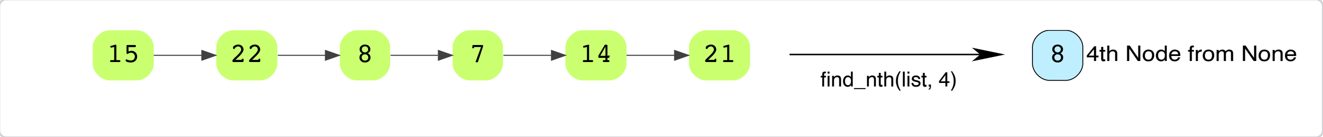
The value of the node *n* positions from the end. Returns `-1` i ou bounds.

Sample Input

```
LinkedList = 22->18->60->78->47->39->99 and n = 3
```

Sample Output

```
47
```



Coding Exercise

Take some time to flesh out the logic for your algorithm. Keep in mind that you need to return the `data` component of the specified node.

This isn't a very tough exercise. All hard work should end on a good note.

Good luck!

main.py

LinkedList.py

Node.py

```
from LinkedList import LinkedList
from Node import Node
# Access head_node => list.get_head()
# Check if list is empty => list.is_empty()
# Delete at head => list.delete_at_head()
# Delete by value => list.delete(value)
# Search for element => list.search()
# Length of the list => list.length()
# Remove duplicates => list.remove_duplicates()
# Node class  {int data ; Node next_element;}

# Function to find the nth node from end of Linked List


def find_nth(lst, n):
    # Write your code here

    ll_len = lst.length()
    if ll_len < n:
        return -1
    else:
        elem_idx = ll_len - n

    lst_curr = lst.get_head()
    for i in range(elem_idx):
        lst_curr = lst_curr.next_element
    return lst_curr.data
```

← Back

Solution Review: Union & Intersection ...

Next →

Solution Review: Return the Nth Node...

✓ Completled

Report an Issue