Write append method

first...

Read over the \_init\_ method for class \_LLNode\_ and \_LinkedList\_.

class LinkedListNode:
    """
    Node to be used in linked list
    
    \_\_\_ Attributes \_\_\_
    next\_ - successor to this LinkedListNode
    value - data represented by this LinkedListNode
    """
    next_: Union["LinkedListNode", None]

    def \_\_init\_\_(self, value: object,
                      next_: Union["LinkedListNode", None]=None) -> None:
        """
        Create LinkedListNode self with data value and successor next
        
        >>> LinkedListNode(5).value
        5
        >>> LinkedListNode(5).next_
        """
        self.value, self._next = value, next_


class LinkedList:
    """
    Collection of LinkedListNodes
    
    \_\_\_ Attributes \_\_\_
    front - first node of this LinkedList
    back - last node of this LinkedList
    size - number of nodes in this LinkedList, \( \geq 0 \)
    """
    front: LinkedListNode
    back: LinkedListNode
    size: int

    def \_\_init\_\_(self) -> None:
        """
        Create an empty linked list.
        """
        self.front, self.back, self.size = None, None, 0

    (continued on next page)
def append(self, value: object) -> None:
    ""
    Insert a new LinkedListNode with value after self.back.
    ""

    >>> lnk = LinkedList()
    >>> lnk.append(5)
    >>> lnk.size
    1
    >>> print(lnk.front)
    5 ->|
    >>> lnk.append(6)
    >>> lnk.size
    2
    >>> print(lnk.front)
    5 -> 6 ->|
    ""
    pass

1. What if this is the first node being appended? Show this with a diagram.

2. What if there are already some nodes in the list? Show this with a diagram.

Now implement the body of append