

next...

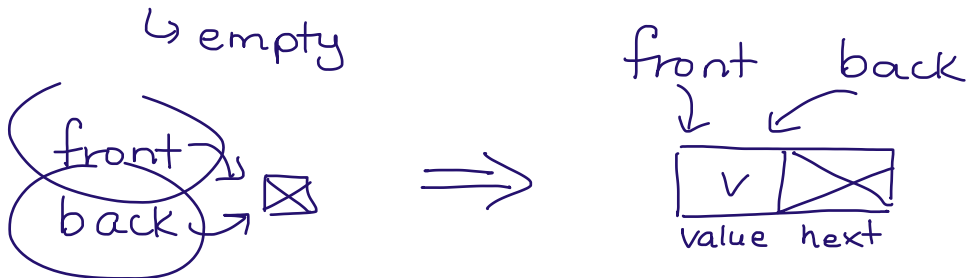
Now, read the header and docstring for the method `append`, and then answer the questions that follow it.

```
def append(self, value):
    """
    Insert a new LinkedListNode with value after self.back.

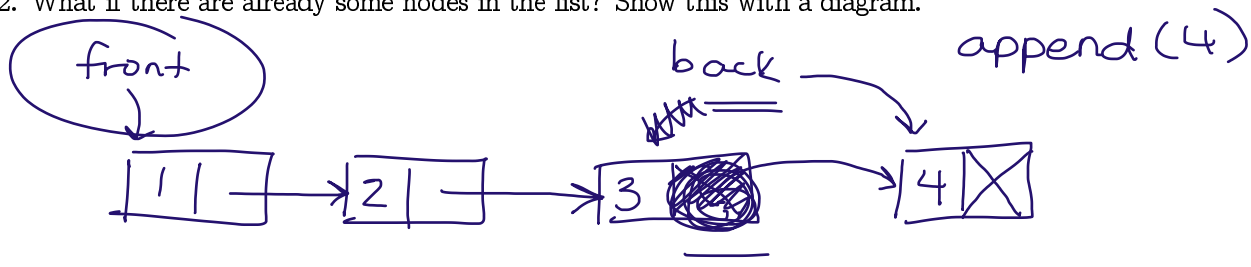
    @param LinkedList self: this LinkedList.
    @param object value: value of new LinkedListNode
    @rtype: None

    >>> 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.



only works in general case

Now implement the body of `append`

- create new node
- change what `back` refers to
- old `back` to refer to new `back`
- if list was empty, update `front`
- increment size

`new_node = LinkedListNode(value)`

`self.back.next = new_node`

`self.back = new_node`

`if self.size == 0:`

`→ self.front = new_node`

`else:`

`self.size += 1`