




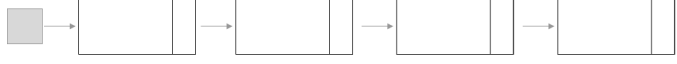
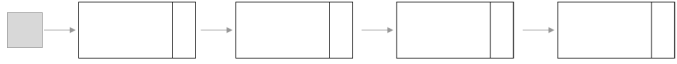
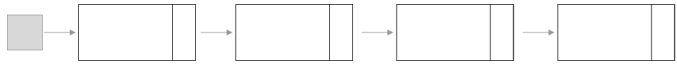
# CSC148 Worksheet 12 Solution

Hyungmo Gu


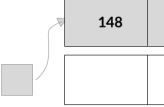

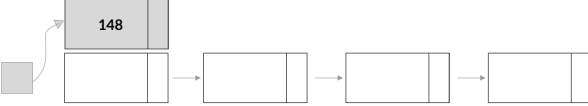
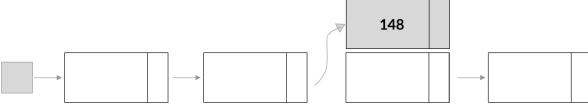

April 26, 2020

## Question 1


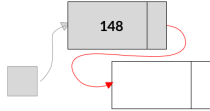

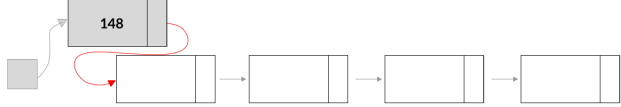
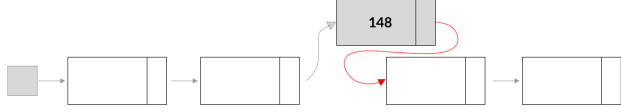

a.

Input description	Linked list diagram
<code>len(self) == 0, index == 0</code>	
<code>len(self) == 1, index == 0</code>	
<code>len(self) == 1, index == 1</code>	
<code>len(self) == 4, index == 0</code>	
<code>len(self) == 4, index == 2</code>	
<code>len(self) == 4, index == 4</code>	

b.

Input description	Linked list diagram
<code>len(self) == 0, index == 0</code>	
<code>len(self) == 1, index == 0</code>	
<code>len(self) == 1, index == 1</code>	
<code>len(self) == 4, index == 0</code>	
<code>len(self) == 4, index == 2</code>	
<code>len(self) == 4, index == 4</code>	

### Correct Solution:

Input description	Linked list diagram
<code>len(self) == 0, index == 0</code>	
<code>len(self) == 1, index == 0</code>	
<code>len(self) == 1, index == 1</code>	
<code>len(self) == 4, index == 0</code>	
<code>len(self) == 4, index == 2</code>	
<code>len(self) == 4, index == 4</code>	

## Question 2

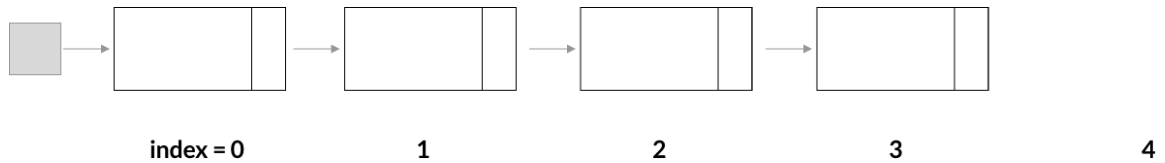
- To reassign `self._first` to something new, `len(self)` can be value, but `index` has to be at 0.
- To make `insert` method to behave the same as `LinkedList.append`, `len(self)` can be any value, but `index = len(self) - 1`.

### Correct Solution:

To make `insert` method to behave the same as `LinkedList.append`, `len(self)` can be any value, but `index = len(self)`.

### Notes:

- Learned that the first node in linked list represents `index = 0`.



- Learned that *insert* operation for linked list is the same as the *insert* operation for lists.
  - If node doesn't exist at this index, then hook last node to the inserting node.
  - If node does exist, then hook current node to one end of the inserting node, and the other end to the next node

## Question 3

```

1  ...
2  def insert(self, index: int, item: Any) -> None:
3      """Insert a the given item at the given index in this list.
4      Raise IndexError if index > len(self) or index < 0.
5      Note that adding to the end of the list is okay.
6
7      >>> lst = LinkedList([1, 2, 3])
8      >>> lst.insert(0,0)
9      >>> str(lst)
10     '[0 -> 1 -> 2 -> 3]'
11     >>> lst.insert(1,10)
12     >>> str(lst)
13     '[0 -> 10 -> 1 -> 2 -> 3]'
14     >>> lst.insert(5,10)
15     >>> str(lst)
16     '[0 -> 10 -> 1 -> 2 -> 3 -> 10]'
17     """
18
19     curr = self._first
20     current_index = 0
21
22     if index == 0:
23         current_node = curr
24         self._first = _Node(item)
25         self._first.next = current_node
26         return
27
28     while curr is not None:
29         if current_index != index - 1:
30             curr = curr.next
31             current_index += 1
32             continue
33
34         # 1. if index == len(self), then append to list
35         # 2. if index < len(self), then insert value at index
36         current_node = curr.next
37         inserting_node = _Node(item)

```

```
38         curr.next = inserting_node
39         inserting_node.next = current_node
40         return
41
42     # 3. if index > len(self), raise IndexError
43     raise IndexError
44     ...
```

Listing 1: worksheet\_12\_q3\_solution.py