Problem # 707: Design Linked List (Medium)

https://leetcode.com/problems/design-linked-

My Solution:

class Node:

def \_\_init\_\_(self, val):

self.val = val

self.next = None

class MyLinkedList:

def \_\_init\_\_(self):

"""

Initialize your data structure here.

"""

self.head = None

self.length = 0

def get(self, index: int) -> int:

"""

Get the value of the index-th node in the linked list. If the index is invalid, return -1.

"""

if index > self.length - 1:

return(-1)

elif index == 0:

return(self.head.val)

#else:

curr = self.head

count = 0

while count < index:

curr = curr.next

count += 1

return(curr.val)

def addAtHead(self, val: int) -> None:

"""

Add a node of value val before the first element of the linked list. After the insertion, the new node will be the first node of the linked list.

"""

new\_node = Node(val)

self.length += 1

if not self.head:

self.head = new\_node

else:

new\_node.next = self.head

self.head = new\_node

def addAtTail(self, val: int) -> None:

"""

Append a node of value val to the last element of the linked list.

"""

new\_node = Node(val)

self.length += 1

if not self.head:

self.head = new\_node

else:

curr = self.head

while curr.next:

curr = curr.next

curr.next = new\_node

def addAtIndex(self, index: int, val: int) -> None:

"""

Add a node of value val before the index-th node in the linked list. If index equals to the length of linked list, the node will be appended to the end of linked list. If index is greater than the length, the node will not be inserted.

"""

if index == 0:

self.addAtHead(val)

elif index == self.length:

self.addAtTail(val)

elif index < self.length:

prev = self.head

curr = self.head.next

count = 1

while count < index:

prev = curr

curr = curr.next

count += 1

new\_node = Node(val)

self.length += 1

prev.next = new\_node

new\_node.next = curr

def deleteAtIndex(self, index: int) -> None:

"""

Delete the index-th node in the linked list, if the index is valid.

"""

if index == 0:

self.head = self.head.next

self.length -= 1

elif index > 0 and index < self.length:

self.length -= 1

prev = self.head

curr = self.head.next

count = 1

while count < index:

prev = curr

curr = curr.next

count += 1

prev.next = curr.next

# Your MyLinkedList object will be instantiated and called as such:

# obj = MyLinkedList()

# param\_1 = obj.get(index)

# obj.addAtHead(val)

# obj.addAtTail(val)

# obj.addAtIndex(index,val)

# obj.deleteAtIndex(index)