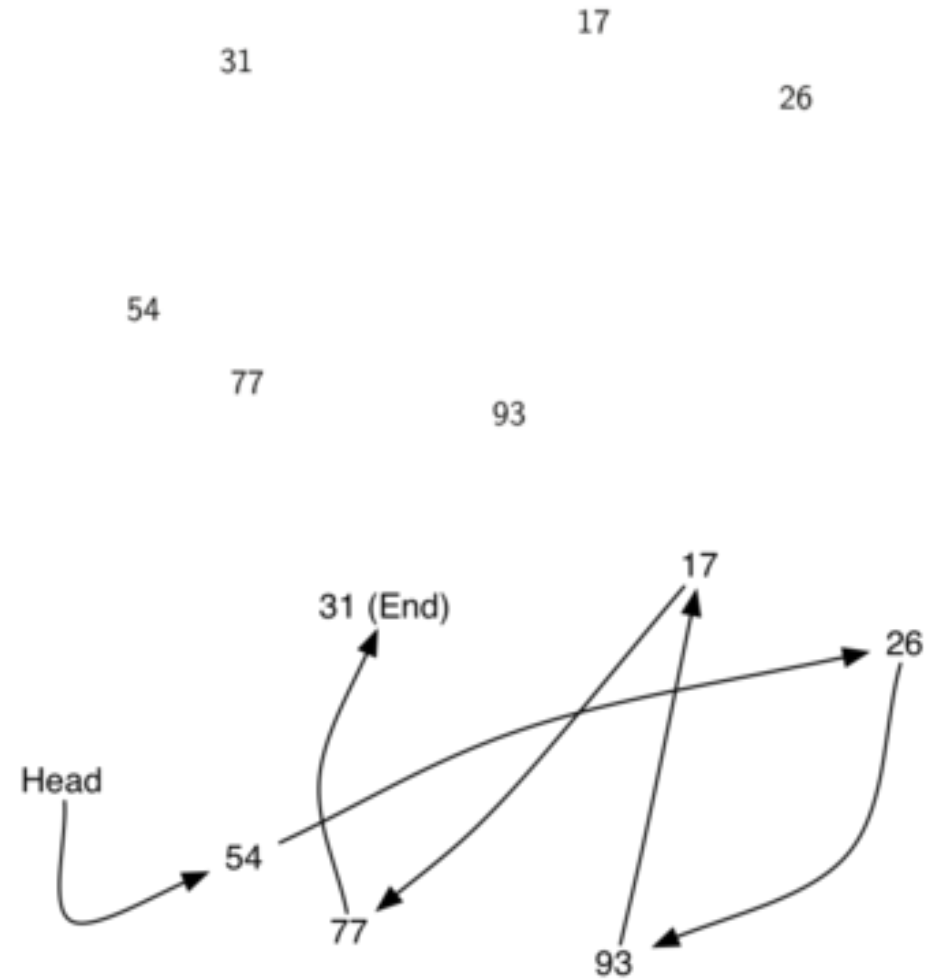


Linked List

List

- Python list
- รายการแบบโยง (linked list)
 - แบบไม่มีลำดับ (unordered list)
 - แบบมีลำดับ (ordered list)
 - น้อยไปมาก
 - มากไปน้อย

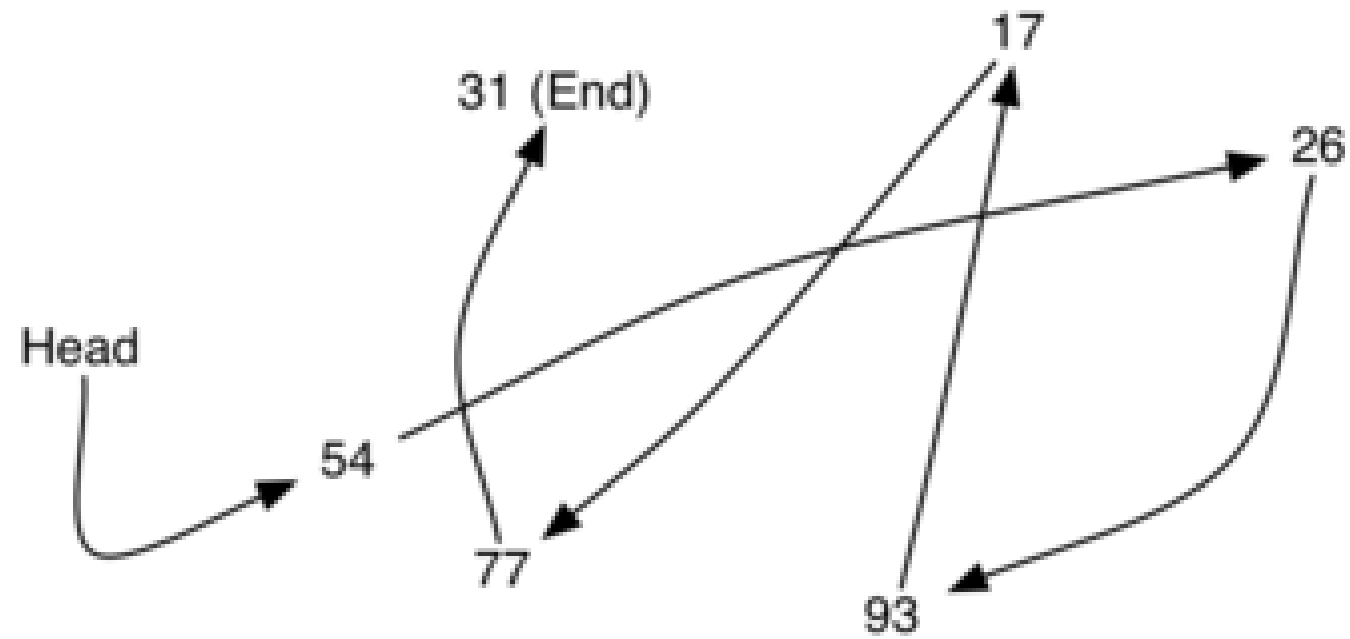


แบบไม่มีลำดับ

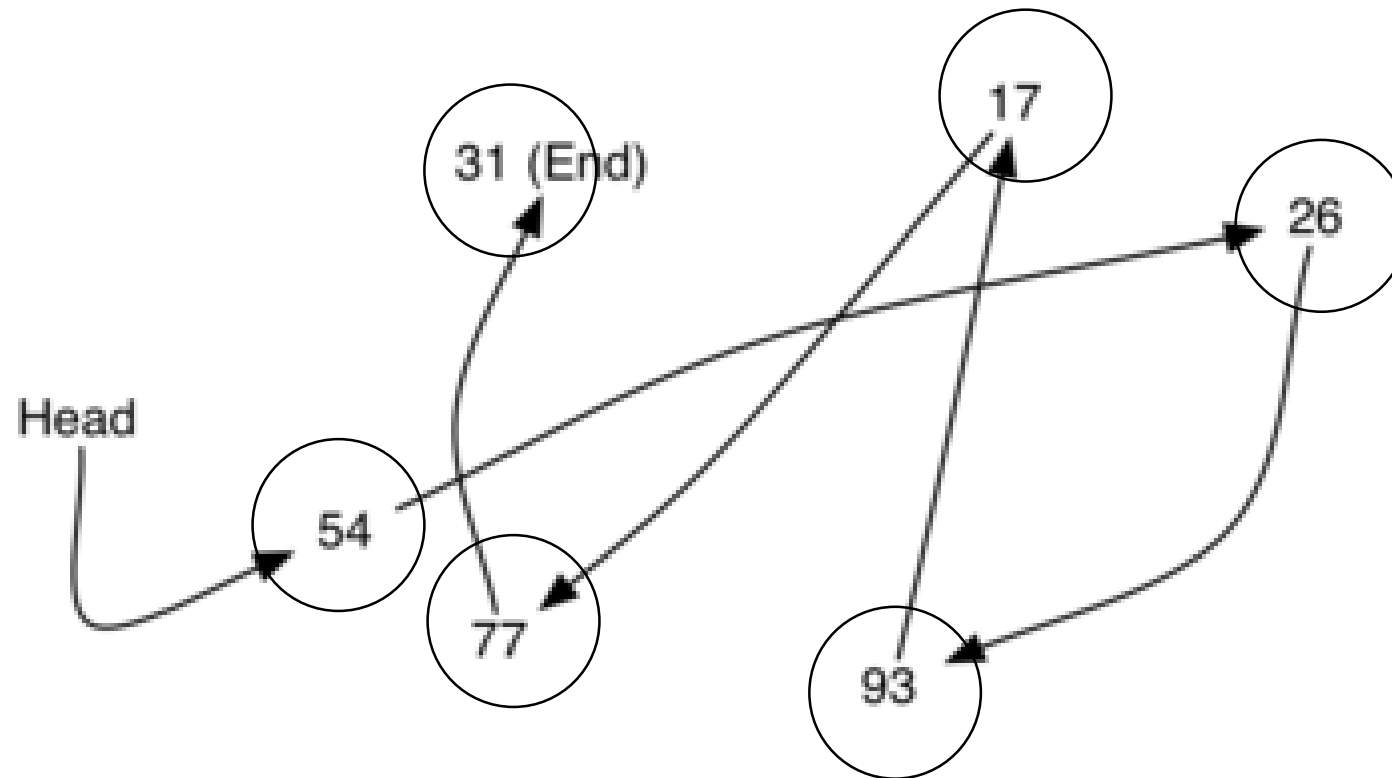
ADT ของ UnorderedList

- `UnorderedList()` ทำการสร้าง `UnorderedList` ใหม่แล้วคืนค่าเป็น `UnorderedList` ว่าง
- `add(item)` รับ parameter คือ `item` เพื่อเพิ่มเข้าใน `UnorderedList`
- `remove(item)` ค้นหา `item` แล้วเอาออกจาก `UnorderedList`
- `search(item)` ค้นหา `item` แล้วคืนค่าว่าเจอหรือไม่ เป็น `True` กับ `False`
- `isEmpty()` ตรวจสอบแล้วคืนค่าว่างหรือไม่
- `size()` คืนค่าขนาดของ `UnorderedList`
- `append(item)` เพิ่ม `item` โดยต่อท้ายใน `UnorderedList`
- `index(item)` หาดัชนีของ `item` ใน `UnorderedList`
- `insert(pos, item)` เพิ่ม `item` ณ ดัชนีที่ `pos`
- `pop()` นำข้อมูลตัวสุดท้ายออกจาก `UnorderedList` และคืนค่านั้นด้วย
- `pop(pos)` นำข้อมูล ณ ดัชนีที่ `pos` ออกจาก `UnorderedList` และคืนค่านั้นด้วย

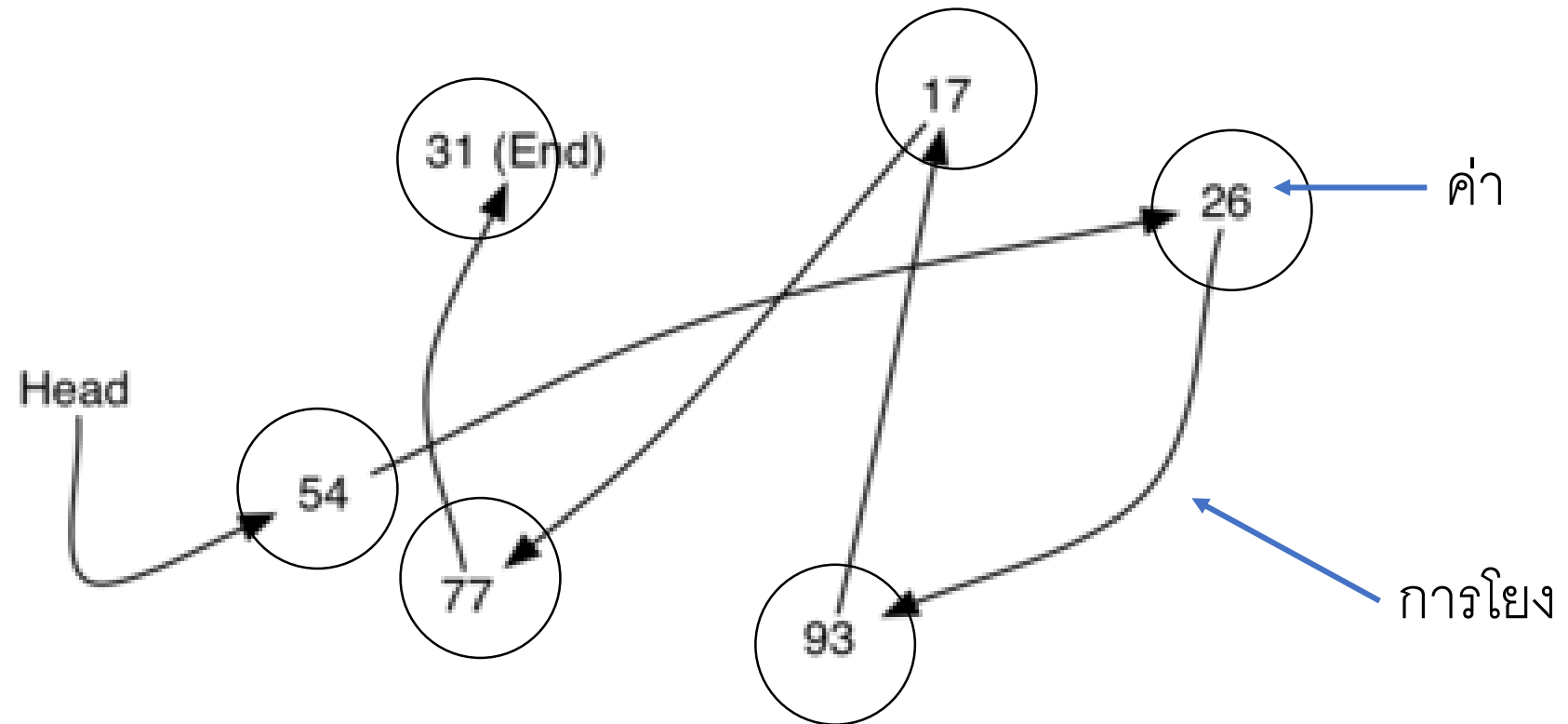
Node



Node



Node



```
class Node:
```

```
    def __init__(self,initdata):
```

```
        self.data = initdata
```

```
        self.next = None
```

```
    def getData(self):
```

```
        return self.data
```

```
    def getNext(self):
```

```
        return self.next
```

```
    def setData(self,newdata):
```

```
        self.data = newdata
```

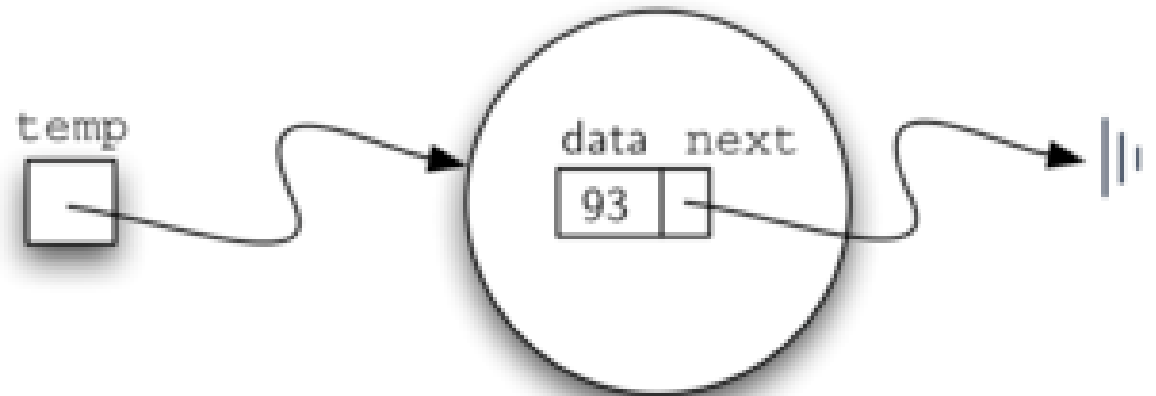
```
    def setNext(self,newnext):
```

```
        self.next = newnext
```

```
>>> temp = Node(93)
```

```
>>> temp.getData()
```

```
93
```



Class ของ UnorderedList

```
class UnorderedList:
```

```
    def __init__(self):  
        self.head = None
```

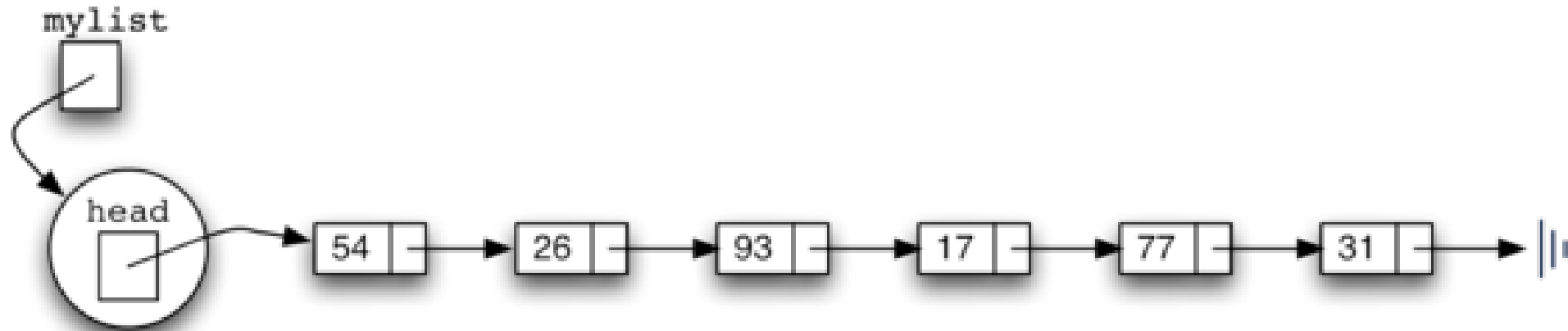
```
    def isEmpty(self):  
        return self.head == None
```

```
>>> mylist = UnorderedList()
```



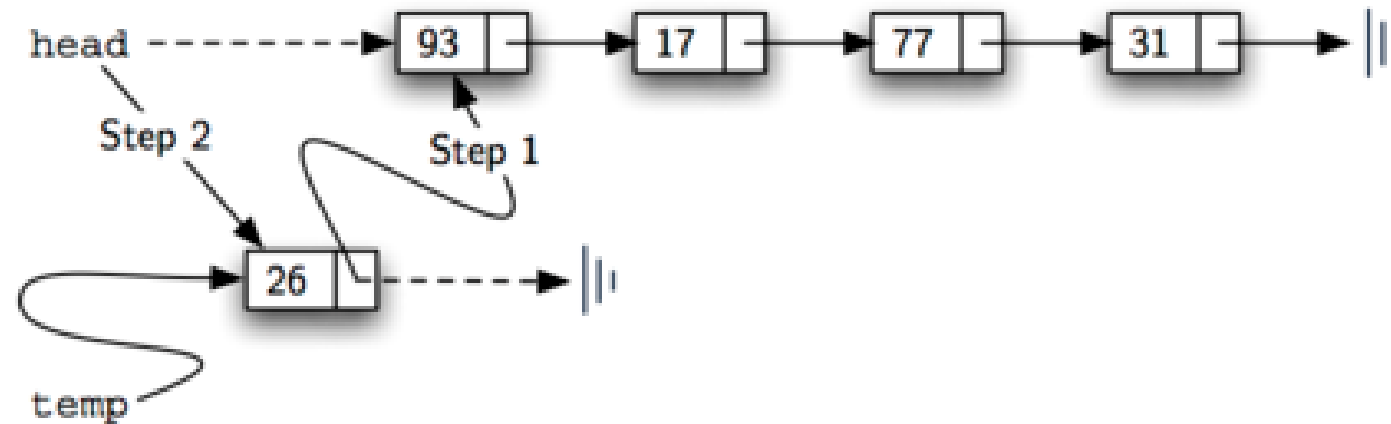
add

```
>>> mylist.add(31)
>>> mylist.add(77)
>>> mylist.add(17)
>>> mylist.add(93)
>>> mylist.add(26)
>>> mylist.add(54)
```



```
def add(self,item):  
    temp = Node(item)  
    temp.setNext(self.head)  
    self.head = temp
```

เพิ่มที่ส่วนหัวของ list

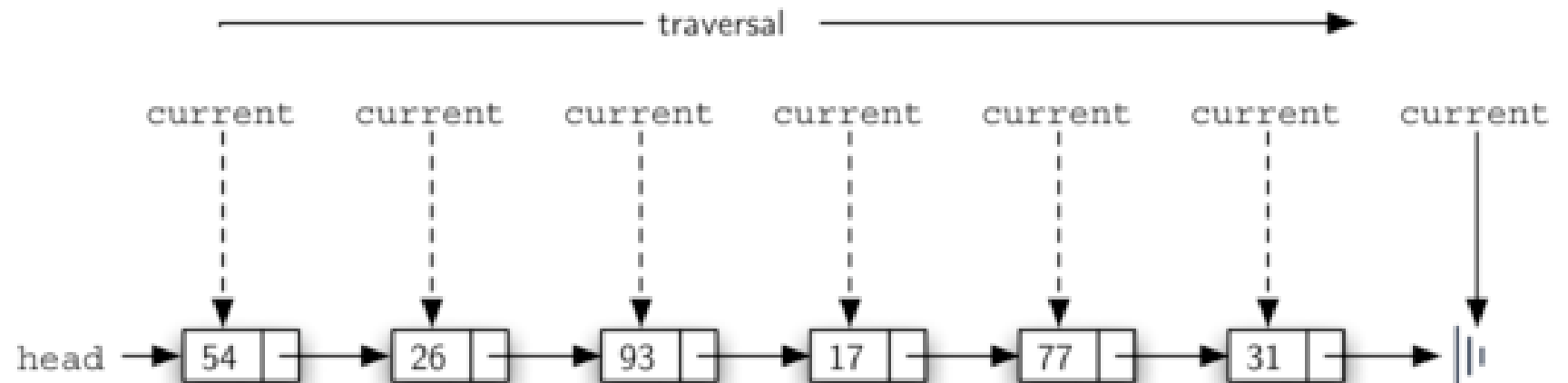


size

```
def size(self):  
    current = self.head  
    count = 0  
    while current != None:  
        count = count + 1  
        current = current.getNext()
```

วิ่งทั่ว list เพื่อนับ

```
return count
```



search

- วิ่งทั่ว **list** จนกว่า
 - เจอ \rightarrow True
 - หมด **list** แล้วไม่เจอ \rightarrow False
- ถูกนำไปดัดแปลงใน **remove**

```
def search(self,item):
```

```
    current = self.head
```

```
    found = False
```

```
    while current != None and not found:
```

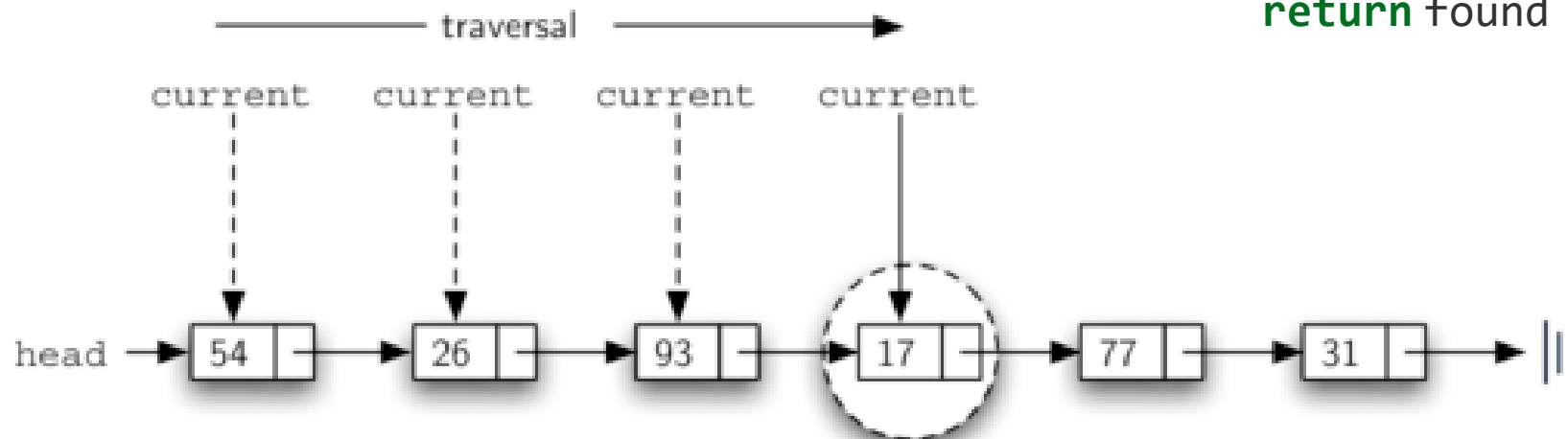
```
        if current.getData() == item:
```

```
            found = True
```

```
        else:
```

```
            current = current.getNext()
```

```
    return found
```



remove

```
def remove(self, item):
```

```
    current = self.head
```

```
    previous = None
```

```
    found = False
```

```
    while not found:
```

```
        if current.getData() == item:
```

```
            found = True
```

```
        else:
```

```
            previous = current
```

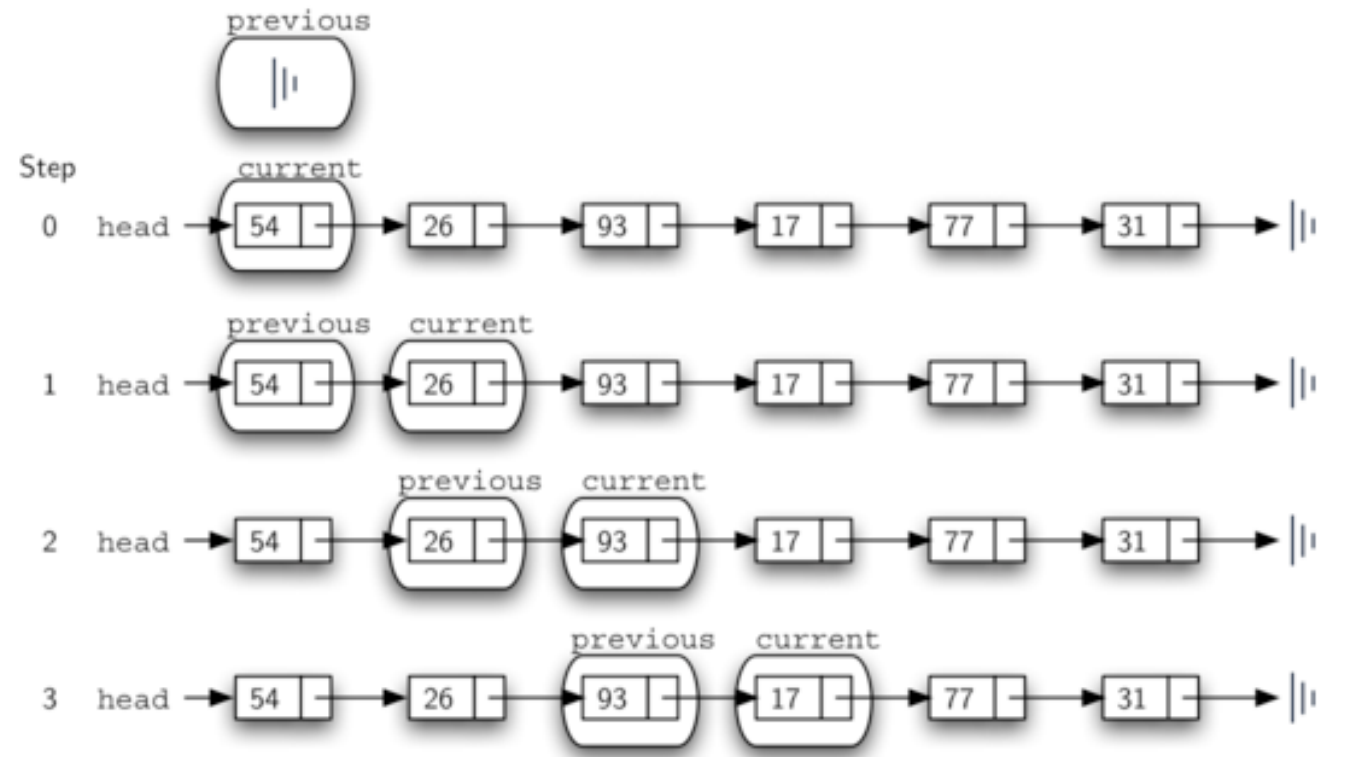
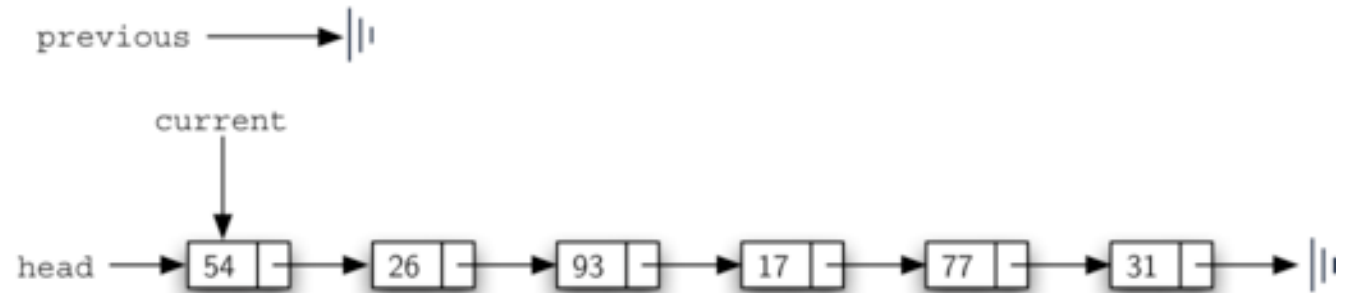
```
            current = current.getNext()
```

```
    if previous == None:
```

```
        self.head = current.getNext()
```

```
    else:
```

```
        previous.setNext(current.getNext())
```



remove

```
def remove(self, item):
```

```
    current = self.head
```

```
    previous = None
```

```
    found = False
```

```
    while not found:
```

```
        if current.getData() == item:
```

```
            found = True
```

```
        else:
```

```
            previous = current
```

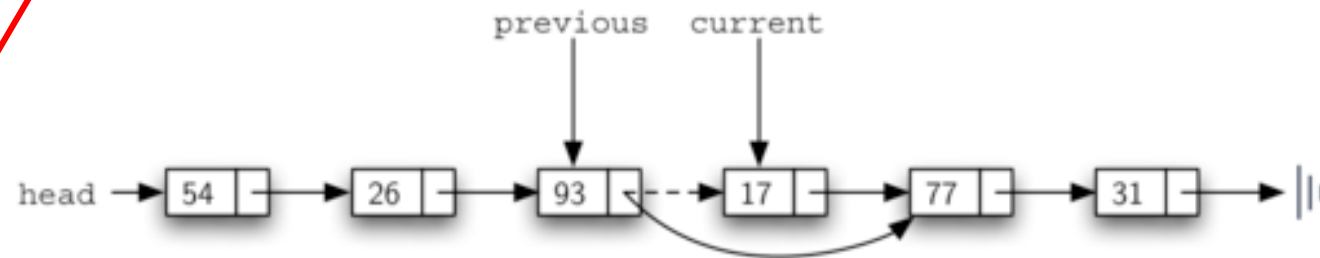
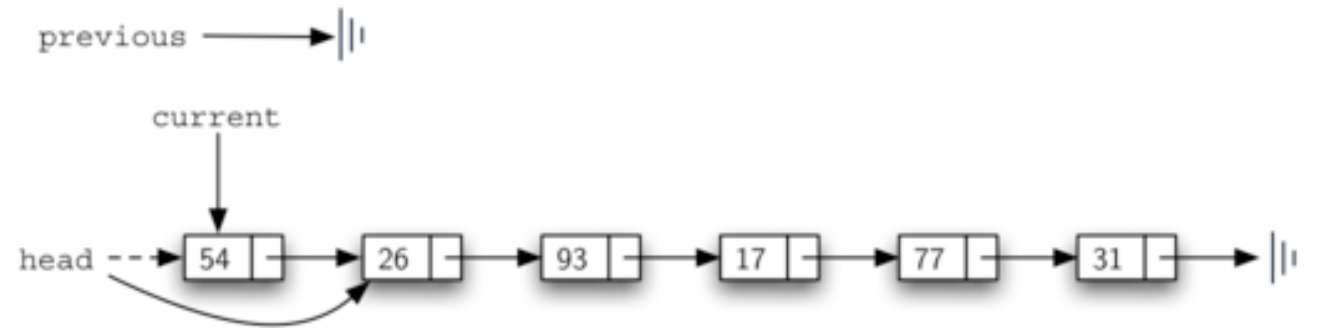
```
            current = current.getNext()
```

```
    if previous == None:
```

```
        self.head = current.getNext()
```

```
    else:
```

```
        previous.setNext(current.getNext())
```



ADT ของ OrderedList

- `OrderedList()` ทำการสร้าง `OrderedList` ใหม่แล้วคืนค่าเป็น `OrderedList` ว่าง
- `add(item)` รับ parameter คือ `item` เพื่อเพิ่มเข้าใน `OrderedList`
- `remove(item)` ค้นหา `item` แล้วเอาออกจาก `OrderedList`
- `search(item)` ค้นหา `item` แล้วคืนค่าว่าเจอหรือไม่ เป็น `True` กับ `False`
- `isEmpty()` ตรวจสอบแล้วคืนค่าว่าว่างหรือไม่
- `size()` คืนค่าขนาดของ `OrderedList`
- `index(item)` หาดัชนีของ `item` ใน `OrderedList`
- `pop()` นำข้อมูลตัวสุดท้ายออกจาก `OrderedList` และคืนค่านั้นด้วย
- `pop(pos)` นำข้อมูล ณ ดัชนีที่ `pos` ออกจาก `OrderedList` และคืนค่านั้นด้วย