Lab_debug Feedback

Average score: 98%

PL average time: 52 minutes

Most but not all found class helpful for completing lab.

Lab generally improved confidence

Lab matched experience levels; request for slightly harder content
Exam 1

Exams will be proctored by the CBTF: https://cbtf.engr.illinois.edu/

(That link will have a link to Prairietest, where you can sign up for exam 1)

Reservations open on February 2nd @ 9 AM

You must take the exam sometime between 2/14 and 2/16!

See website for expected content:

https://courses.grainger.illinois.edu/cs277/sp2023/exams/
Learning Objectives

- Conceptualize and code a linked list
- Conceptualize an array list
- Compare list implementations
Linked Lists

class Node:
    def __init__(self, data, next=None):
        self.data = data
        self.next = next

class linkedList:
    def __init__(self, head=None):
        self.head = head
Linked List Structure (Node vs List)

```python
n1 = Node(3)
n2 = Node(5)
n3 = Node(7)
n1.next = n2
n2.next = n3
```

```python
t1 = [3, None]
t2 = [5, None]
t3 = [7, None]
t1[1]=t2
t2[1]=t3
```

```python
curr = n1
print(curr.next.next.data)
curr = t1
print(t1[1][1][0])
```
Class Linked List

class Node:
    def __init__(self, data, next=None):
        self.data = data
        self.next = next

class linkedList:
    def __init__(self, head=None):
        self.head = head

    def __str__(self):
    def __len__(self):
    def __getitem__(self):

    def add(self):
    def insert(self):
    def delete(self):
    def remove(self):
Linked List Add

```python
ll = linkedList()
for i in range(3):
    ll.add(i)
```
## Linked List Add

```python
def add(self, data):
    temp = self.head
    self.head = Node(data, temp)
```

```python
ll = linkedList()
for i in range(3):
    ll.add(i)
```
```
ll = linkedList()
for i in range(3):
    ll.add(i)
print(len(ll))
```
Linked List Length

```python
class linkedList:
    def __len__(self):
        i = 0
        curr = self.head
        while(curr):
            curr = curr.next
            i+=1
        return i

ll = linkedList()
for i in range(3):
    ll.add(i)
print(len(ll))
```

```
head
None
```

```
0 1 2
```
1. \texttt{ll = linkedList()}
2. for \texttt{i in range(3)}:
3. \hspace{1em} \texttt{ll.add(i)}
4. \texttt{print(ll)}

```python
class linkedList:
    def \texttt{__str__}(self):
        curr = self.head
        out="["
        while(curr):
            out+="{},".format(curr.data)
            curr = curr.next
        if out[-1]==",":
            out = out[:-1]
        out = out +"]n"
        return out
```

```
head
0  1  2  None
```
Linked List `__getitem__()`

```python
ll = linkedList()
for i in range(5):
    ll.add(i)
print(ll[3])
```

```
head
```

```
0   1   2   3   4
```

```
None
```
Linked List `__getitem__()`

```python
ll = linkedList()
for i in range(5):
    ll.add(i)
print(ll[3].data)
```

```python
def __getitem__(self, pos):
    curr = self.head
    i = 0
    while(curr and i < pos):
        curr = curr.next
        i+=1
    if i == pos:
        return curr
    else:
        raise ValueError("Out of bounds")
    return None
```

```
head
```

```
0 -> 1 -> 2 -> 3 -> 4 -> None
```
for i in range(5):
    ll.add(i)
ll.insert("Value", 2)
print(ll)
Linked List insert()

```python
for i in range(5):
    ll.add(i)
ll.insert("Value", 2)
print(ll)
```

```
None
7
6
5
4
3
2
1
0
head
```
Linked List insert()

```python
for i in range(5):
    ll.add(i)

ll.insert("Value", 2)
print(ll)
```

```python
def insert(self, data, pos=0):
    if (pos == 0):
        self.add(data)
    else:
        prev = self.__getitem__(pos-1)
        temp = prev.next
        prev.next = Node(data,temp)
```

```
head

0 -> 1 -> 2 -> 3 -> 4 -> None
```
Linked List delete()

```python
for i in range(5):
    ll.add(i)
ll.delete(1)
print(ll)
```

```
0 1 2 3 4
```

head

```
None
```
Linked List delete()

```python
for i in range(5):
    ll.add(i)
ll.delete(1)
print(ll)
```

```
0 1 2 3 4
None
```
Linked List delete()

```python
for i in range(5):
    ll.add(i)
ll.delete(1)
print(ll)
```

```python
def delete(self, i):
    if i == 0:
        self.head = self.head.next
    else:
        prev = self.__getitem__(i-1)
        prev.next = prev.next.next
```
In-Class Exercise: remove()

```python
ll = linkedList()
for i in range(5):
    ll.add(i)
    ll.add(i)
ll.remove(4)
```

![Linked List Diagram]
In-Class Exercise: find()

```python
#initialize ll
node = ll.find("B")
if(node):
    print("Exists!")
else:
    print("Doesnt exist!")
```

head

```
0 -> 1 -> 2 -> 3 -> 4 -> None
```
Linked List Pros and Cons
What do we care about when we write code?