第4次上机作业

姜城

2015/04/20

第一题

- append

```
def append(self, item):
    newNode=Node(item)
    current=self.head
    if current==None:
        self.head=newNode
        return
    while current.getNext()!=None:
        current=current.getNext()
    current.setNext(newNode)
```

第一题

- index

```
def index(self, item):
    current=self.head
    count=0
    while current!=None:
        if current.getData()==item:
            return count
        count+=1
        current=current.getNext()
    raise ValueError("%s not in list" % item)
```

第一題

- pop

```
def pop(self, index=None):
    index=(self.size()-1) if index==None else index
    current=self.head
    previous=None
    counter=0
    while counter<index and current!=None:
        previous=current
        current=current.getNext()
        counter+=1
    if current==None:
        raise IndexError("Index out of range")
    elif previous==None:
        self.head=current.getNext()
    else:
        previous.setNext(current.getNext())
    return current.getData()
```

第一题

- insert

```
def insert(self, position, item):
    newItem=Node(item)
    current=self.head
    previous=None
    count=0
    while count<position and current!=None:
        previous=current
        current=current.getNext()
        count+=1
    if previous==None:
        newItem.setNext(current)
        self.head=newItem
    else:
        previous.setNext(newItem)
        newItem.setNext(current)
```

第一題

- __str__

```
def __str__(self):
    if self.head==None:
        return "[]"
    s="["
    current=self.head
    while current!=None:
        if isinstance(current.getData(), str):
            s+="'"+current.getData()+"'"+", "
        else:
            s+=str(current.getData())+", "
        current=current.getNext()
    s=s[0:len(s)-2]+"]"
    return s
```

第一题

```
def __getslice__(self, i, j):
    newList=UnorderedList()
    if j<=i:
        return newList
    count=0
    current=self.head
    while current!=None:
        if i<=count<j:
            newList.append(current.getData())
        elif count>=j:
            break
        else:
            pass
        current=current.getNext()
        count+=1
    return newList

def __len__(self):
    '''
    when the parameters of __getslice__() are negative, __len__() will be called
    '''
    return self.size()
```

```
def __getitem__(self, key):
    if isinstance(key, int):
        size=self.size()
        key=key+size if key<0 else key
        if key<0 or key>=size:
            raise IndexError("out of range")
        count=0
        current=self.head
        while count<key:
            current=current.getNext()
            count+=1
        return current.getData()
    elif isinstance(key, slice):
        (begin, end, step)=key.indices(self.size())
        newList=UnorderedList()
        count=0
        current=self.head
        while current!=None:
            if begin<=count<end and (count-begin)%step==0:
                newList.append(current.getData())
            elif count>=end:
                break
            else:
                pass
            current=current.getNext()
            count+=1
        return newList
    else:
        pass
```

第二题

```
class OrderedList(UnorderedList):
    def __init__(self):
        UnorderedList.__init__(self)
```

第二题

```
def add(self, item):
    newNode=Node(item)
    current=self.head
    previous=None
    while current!=None and current.getData()<=item:
        previous=current
        current=current.getNext()
    if previous==None:
        self.head=newNode
        newNode.setNext(current)
    else:
        previous.setNext(newNode)
        newNode.setNext(current)
```

第二题

```
def remove(self, item):
    current=self.head
    previous=None
    while current!=None and current.getData()<item:
        previous=current
        current=current.getNext()
    if current!=None and current.getData()==item:
        if previous==None:
            self.head=current.getNext()
        else:
            previous.setNext(current.getNext())
    else:
        raise ValueError("value not in list")
```

第二题

```
def search(self, item):
    current=self.head
    while current!=None and current.getData()<item:
        current=current.getNext()
    if current!=None and current.getData()==item:
        return True
    else:
        return False
```

第三题

```
class Stack():
    def __init__(self):
        self.list=UnorderedList()

    def isEmpty(self):
        return self.list.isEmpty()

    def push(self, item):
        #self.list.append(item) # O(n)
        self.list.add(item) # O(1)

    def pop(self):
        #return self.list.pop() # O(n)
        self.list.pop(0) # O(1)

    def peek(self):
        #return self.list[-1:].pop() # O(n)
        return self.list[0:1].pop() # O(1)

    def size(self):
        return self.list.size()
```

第三题

```
class Queue():
    def __init__(self):
        self.list=UnorderedList()

    def isEmpty(self):
        return self.isEmpty()

    def enqueue(self, item):
        self.list.add(item)

    def dequeue(self):
        return self.list.pop()

    def size(self):
        return self.list.size()
```

第四题

```
class DNode():
    def __init__(self, initData):
        self.data=initData
        self.next=None
        self.prev=None
    def getData(self):
        return self.data
    def getNext(self):
        return self.next
    def getPre(self):
        return self.prev
    def setData(self, newData):
        self.data=newData
    def setNext(self, newNext):
        self.next=newNext
    def setPre(self, newPre):
        self.prev=newPre
```

第四题

```
class DUnorderedList():
    def __init__(self):
        self.head=None
        self.tail=None
```

```
def add(self, item):
    newNode=DNode(item)
    if self.head!=None:
        self.head.setPrev(newNode)
        newNode.setNext(self.head)
    self.head=newNode

    if self.tail==None:
        self.tail=newNode
```

第四题

```
def append(self, item):
    newNode=DNode(item)
    if self.tail!=None:
        self.tail.setNext(newNode)
        newNode.setPrev(self.tail)
    self.tail=newNode

    if self.head==None:
        self.head=newNode
```

第四题

```
def remove(self, item):
    current=self.head
    while current!=None and current.getData()!=item:
        current=current.getNext()
    if current==None:
        raise ValueError("value not in list")
    else:
        prevc=current.getPrev()
        nextc=current.getNext()
        if prevc!=None:
            prevc.setNext(nextc)
        else:
            self.head=nextc
        if nextc!=None:
            nextc.setPrev(prevc)
        else:
            self.tail=prevc
```

第四题

```
def insert(self, position, item):
    current=self.head
    count=0
    while count<position and current!=None:
        current=current.getNext()
        count+=1
    if current==None:
        self.append(item)
    else:
        newItem=DNode(item)
        prev=current.getPrev()
        newItem.setPrev(prev)
        newItem.setNext(current)
        current.setPrev(newItem)
        if prev==None:
            self.head=newItem
        else:
            prev.setNext(newItem)
```

随堂练习

```
def fact(n):
    if n==0:
        return 1
    return n*fact(n-1)

def revstring(s):
    if len(s)==1:
        return s
    else:
        return s[-1]+revstring(s[0:-1])

def palcheck(s):
    if len(s)<=1:
        return True
    elif s[0]!=s[-1]:
        return False
    return palcheck(s[1:-1])
```