Data Structures

Homework Assignment 5 - Linked List - DLL

The total score for this assignment is 50 Points. The other 50 points are awarded for solving the assignment LinkedList - SLL, released before the midterm exam.

Problem 1 – Reverse DoublyLinkedList - 25 Points
Problem 2 – Rotate Doubly Linked List - 25 Points

25% of Gradescope Autograder test cases are hidden for this assignment.

Spring 2024 - Revision 1 - *November*

Problem 1 – Reverse DoublyLinkedList- 50 Points

Implement the member function `reverse()`, which reverses a DoublyLinkedList.

Example

```
ls1 = DoublyLinkedList()
ls1.insertAtFirst(4)
ls1.insertAtFirst(2)
ls1.insertAtFirst(2)
ls1.insertAtFirst(1)

print(ls1)  # Should print: Header-->1-->2-->2-->4-->Trailer
ls1.reverse()
print(ls1)  # Should print: Header-->4-->2-->2-->1-->Trailer
```

Requirements

- Your function has to be in O(n) time complexity.
- Your function has to be in O(1) space complexity.
- You can not change or replace node elements.
- Your function has to work in place.

Problem 2 – Rotate Doubly Linked List - 50 Points

Implement the member function `rotate(n)`, which rotates the DoublyLinkedList to the right by `n` nodes. After the rotation, the `nth` node from the end becomes the new head.

Example

```
dll4 = DoublyLinkedList()
dll4.insertAtEnd(1)
dll4.insertAtEnd(2)
dll4.insertAtEnd(3)
dll4.insertAtEnd(4)
dll4.insertAtEnd(5)

print(dll4)  # Should print: Header-->1<-->2<-->3<-->4<-->5-->Trailer
dll4.rotate(2)
print(dll4)  # Should print: Header-->4<-->5<-->1<-->2<-->3-->Trailer
```

Requirements

- Your function has to be in O(n) time complexity.
- Your function has to be in O(1) space complexity.
- You can not change or replace node elements.