Data Structures and Algorithms

SE-F22

Assignment **Issue Date: May 11, 2024**

Total Marks: 60

Objective:

The objective of this assignment is to solidify our understanding of linked lists.

Instructions:

1) Follow the question instructions very carefully, no changes in function prototypes are allowed.
2) Anyone caught in an act of plagiarism would be awarded an “F” grade in this Lab.

Task 01(Circular Doubly Linked List (CDLL)) [40 Marks]

1) You are not allowed to use tail pointer 2) Boundary checks are crucial for marks.

```
template<typename T>
struct Node
{
        T data;
        Node<T>* next;
        Node<T>* prev;

        Node();
        Node(T val);
}; (5)

template<typename T>
class CDLL
{
        Node<T>* head;
public:
        CDLL(); // Constructor (2)
        ~CDLL(); //Destructor (2)
        CDLL<T>(const CDLL<T>& ref) //Copy Constructor (3)
        void insertAtHead(T val); // Insert val at head (2)
        void insertAtTail(T val); //Insert val at tail (2)
        void insertAfter(T key, T val); // insert val after first occurrence of key (3)
        void insertBefore(T key, T val); // insert val before first occurrence of key (3)
        void removeAtHead(); // Remove Node at Head (3)
        void removeAtTail(); // Remove Node at Tail (3)
        void removeAfter(T key); // Remove the node after first occurrence of key (3)
        void removeBefore(T key); //Remove node before first occurrence of key (3)
        void remove(T key); // Remove first occurrence of key (3)
        void display(); (3)

//Sample output:
        prev: 3 ; data: 1 ; next: 2
        prev: 1 ; data: 2 ; next: 3
        prev: 2 ; data: 3 ; next: 1
 }; (35)
```

Task 02(Rotate CDLL) [30 Marks]

Implement this function as member function of previous class.
Using the CDLL class you created, if you are given a sorted linked list:

```
loop to end <- 1 <-> 2 <-> 3 <-> 4 <-> 5 ->loop to start
void rotateCircularDLL(int times);

rotateCircularDLL(2);
loop to end <- 4 <-> 5 <-> 1 <-> 2 <-> 3 ->loop to start

rotateCircularDLL(4);
loop to end <- 2 <-> 3 <-> 4 <-> 5 <-> 1 ->loop to start
```

You are given a circular doubly linked list containing integer elements. Implement a function to rotate the list to the right by a specified number of positions.

If the times parameter is greater than length of linked list, rotate it: times MODULO length of list.

Example: `loop to end <- 1 <-> 2 <-> 3 <-> 4 <-> 5 ->loop to start`
Length: 5
times: 7

7 % 5 = 2 (rotate 2 times)

Task 03(Find Target Sum) [30 Marks]

Implement this function as member function of previous class.

Using the CDLL class you created, if you are given a sorted linked list:

```
prev: 5 ; data: 1 ; next: 2
prev: 1 ; data: 2 ; next: 3
prev: 2 ; data: 3 ; next: 4
prev: 3 ; data: 4 ; next: 5
prev: 4 ; data: 5 ; next: 1

and a target sum, you have to check if there exists two nodes, that sum up to the target.

bool targetSum(int target);
```

targetSum(3) => true (1 + 2)
targetSum(5) => true (1 + 4) or (2 + 3)
targetSum(10) => false

Note: You are required to write a O(n) time complexity solution.

Khuda ke liyai, khud try kr lena ☺