

CSE211 DATA STRUCTURES

LAB 1 FALL 2024

LINKED LIST

Prerequisites

Open the Ubuntu terminal and type the following after downloading the `tarball` file:

```
cd /mnt/c/Users/user/Downloads
tar -xvf lab1_1.tar.gz -C /mnt/c/Users/user/Desktop/lab1
cd /mnt/c/Users/user/Desktop/lab1 && make all
code .
```

Introduction

In this lab, you will implement various operations on a singly linked list using C++. The linked list is implemented as a template class `LinkedList` that stores elements of type `T`. Your task is to complete the implementation of the following functions in the `LinkedList` class:

1. `reverseInGroups`: Reverse the linked list in groups of K nodes.
2. `findKthFromEnd`: Find the Kth node from the end of the linked list in one pass.
3. `detectAndRemoveLoop`: Detect if there is a loop in the linked list and remove it if it exists.
4. `print`: Print the elements of the linked list.

Project Structure

The project has the following structure:

```
.
├── bin // executable files will be generated here by the compiler
│   └── linkedlist
├── include // header files
│   └── LinkedList.hpp
├── obj // object files will be generated here by the compiler
│   ├── LinkedList.o
│   └── main.o
├── src // source files
│   ├── LinkedList.cpp
│   └── main.cpp
├── instructions.pdf // instructions for the lab
└── Makefile // Makefile to build and run the project
```

- `include/LinkedList.hpp`: Header file containing the declaration of the `LinkedList` class and its member functions.
- `src/LinkedList.cpp`: Source file where you will implement the member functions of the `LinkedList` class.
- `src/main.cpp`: Main source file that demonstrates the usage of the `LinkedList` class.

- `Makefile`: Makefile to build and run the project.

Instructions

1. Navigate to the `src/LinkedList.cpp` file.
2. Implement the following functions in the `LinkedList` class:
 - `reverseInGroups(std::size_t k)`: Reverse the linked list in groups of K nodes.
 - `findKthFromEnd(std::size_t k)`: Find the Kth node from the end of the linked list in one pass.
 - `detectAndRemoveLoop()`: Detect if there is a loop in the linked list and remove it if it exists.
 - `print()`: Print the elements of the linked list.
3. Refer to the function documentation comments in the `include/LinkedList.hpp` file for more details on each function.
4. You can use the `main.cpp` file to test your implementation manually.
5. To compile and run the project, use the provided `Makefile` by running the following commands in the terminal:

```
make all
```

6. To run the test cases, use the following command:

```
make run
```

Make sure all the main cases pass successfully.

Hints

- For `reverseInGroups`, you can use a similar approach as reversing the entire linked list, but limit the reversal to K nodes at a time.
- For `findKthFromEnd`, you can use the two-pointer approach. Move one pointer K nodes ahead of the other pointer, then move both pointers simultaneously until the first pointer reaches the end. The second pointer will be at the Kth node from the end.
- For `detectAndRemoveLoop`, you can use Floyd's cycle-finding algorithm (also known as the "tortoise and hare" algorithm) to detect the loop. Once the loop is detected, you can find the starting point of the loop and remove it by setting the next pointer of the last node in the loop to `nullptr`.
- For `print`, you can traverse the linked list and print the elements in the desired format.

Submission

Once you have completed the implementation and tested it thoroughly, submit your `LinkedList.cpp` file.

Evaluation

Your implementation will be evaluated based on the following criteria:

- Correctness: The functions should produce the expected output for various test cases.
- Efficiency: The functions should have an optimal time complexity.
- Code quality: The code should be well-organized, readable, and follow good programming practices.

Warning

- Do not modify any other files except `LinkedList.cpp`.
- Implement only the specified functions.
- Do not add any additional files or directories.
- Do not use any global variables.
- Do not use any additional libraries or imports.
- Any **cheating** will result in a **0** score for the lab.

Good luck with the lab!