Time it Took Matthew

2 hours

What to Submit

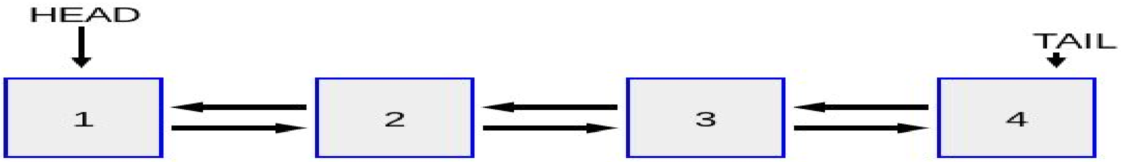
A **FOLDER** named DoublyLinkedList that contains

- A modified version of the CMakeLists.txt that you were given to create an interface library
  - This CMakeLists.txt is in the folder DoublyLinkedList
- DoublyLinkedList.h
- Any other .h files that make up your solution
  - You will need ones for forward iterators, const forward iterators, reverse iterators, const reverse iterators, and a node.
  - An example layout has been given to you

Problem Description

You are to implement a templated doubly linked list class and both forward and reverse iterators over it. The .h files you have been given list what functions in each class you need to implement and what they are supposed to do. You are of course free to add more members and methods if you would like.

A doubly linked list is made up of nodes and pointers to the first element (head) and a pointer to the last element(tail). Each node contains a value, a pointer to the previous element in the list, and a pointer to the next element in the list. An image of a doubly linked list is below.



Google Test

This project makes use of Google Test. Slides for how to install Google Test can be found here. A video walkthrough can be found here.

Hints

- Be careful when doing things with the empty list and lists that only have a single item or operations that reduce the list to a single element
  - You will have to make sure to update head and tail correctly
- A lot of the iterator work is the same and repetitive. Take a look at the StringClassGenericIterator.h for how to create both const and nonconst iterators without having to copy and paste code around. I got inspiration for how to do this from this Stack Overflow post.
  - If you want to copy and paste for the const and nonconst version that is fine
- To get your program to compile as soon as possible you can create functions that do nothing.
  - This will allow you to then add in features and test your features as you add them
- The testing of your code relies heavily on the iterators so make sure that they are working