

Doubly Linked List

Matthew's Stats

Time: 2 hours

Files: 4

Lines of Code: 641

What to Submit

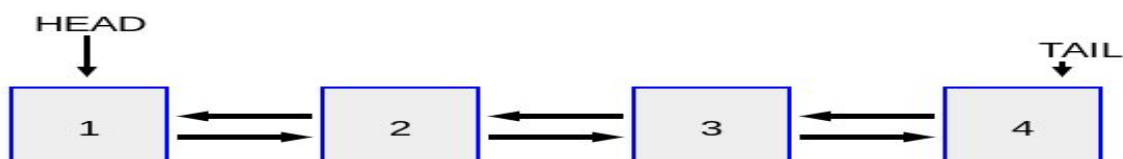
A zip file containing

- 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
- A modified version of the CMakeLists.txt that you were given to create an interface library
 - This CMakeLists.txt is in the folder DoublyLinkedList

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.



Input

- Input will come through GoogleTest so you don't have to worry about taking in input from the command line or standard input.

Google Test

We will be using [Google Test](#) to test your code. The CMakeLists.txt that I have provided you will download Google Test for you and build it as part of your project so you don't need to do anything to get it.

To run the tester after you have built your project do the following

- CLion: Select DoubleLinkedListTester as the target to run. This can be done by changing the value in this box to Tester



- Doing it from the command line you should find DoubleLinkedListTester under your build_folder.

Build times for Google Test will be quite a bit longer than previous assignments due to all of the templated code in Google Test.

You will need to implement all of the specified functions before you can begin testing. You can provide some default return values for the functions you haven't completed yet to be able to test the ones that you have finished. For example, you could have all of the matrix functions return a matrix that contains only the value 0 if you haven't finished it yet to be able to test your vector functions.

Hints

- Be careful when doing things with the empty list, lists that only have a single item, operations that reduce the list to a single element, and operations that make the list empty
 - You will have to make sure to update head and tail correctly in these cases
- 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 and probably easier for you to do
- 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