

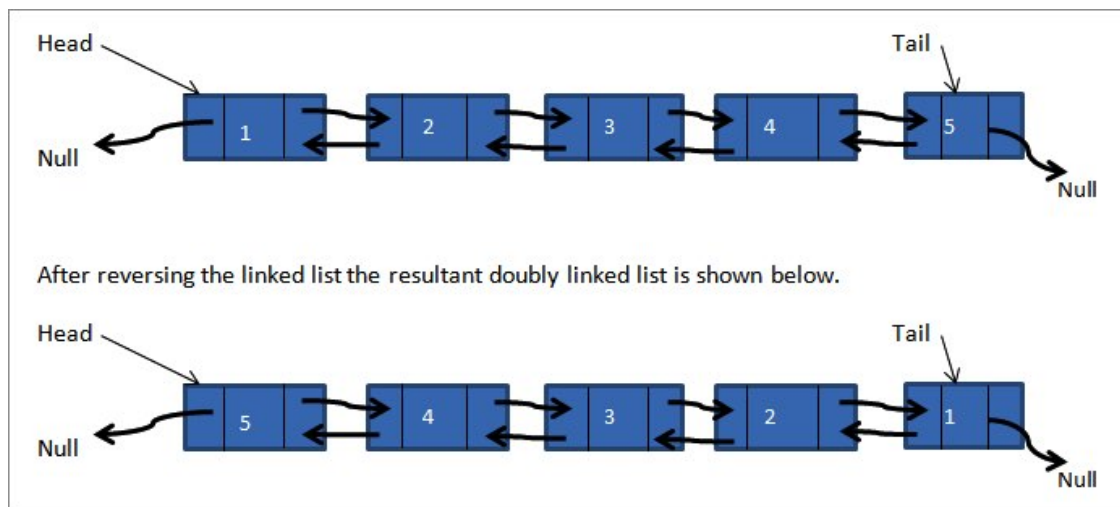
Lab Assignment: Doubly Linked List

In this assignment, we are going to implement a number of methods/algorithms to complete the DoublyLinkedList data structure. Use C++ programming language to complete this assignment.

Deliverable 1: First, use the DoublyLinkedList data structure that has been discussed in the lecture and update it as a template class. After this implementation, it should be possible to declare a DoublyLinkedList of a specific type, such as `DoublyLinkedList<string> city_names;`

Deliverable 2: Define and implement a copy constructor for the DoublyLinkedList class.

Deliverable 3: Implement a `deleteItem` algorithm in the DoublyLinkedList class so that when an item is passed as parameter in the `deleteItem` method, all instances of the item would be removed from the DoublyLinkedList variable.



Deliverable 4: Further, optimize the `delete(index)` method. When the index position is closer from the begin, navigate the doubly linked list starting from the head. However, when the index position is closer from the end, navigate to the index position by using the tail pointer.

For example, if we would like to `delete(4)` then your program should start navigating from tail to reach the index 4 and delete the element at index position 4.

Determine the Big O of the updated algorithm.

Deliverable 5: In this method implementation in the DoublyLinkedList class, reverses the content of the doubly linked list as shown in the figure above. In this scheme, in the loop create two pointers. One pointer will point to head named `hNode` and another will point to tail named `tNode`. Now, exchange the values of the `hNode` and `tNode`. In the next iteration, move `hNode` to point to the next Node and move `tNode` to point to the previous node. Repeat the steps to exchange the values between these two nodes.

Deliverable 6: Update the search(searchItem) algorithm in the DoublyLinkedList class and return all index positions of the searchItem (if there are duplicates). The return value would be a vector variable that will store all the index positions of the searchItem in the DoublyLinkedList variable.

Deliverable 7: Define and implement all the methods of the following Student class. By using the overloaded operator == it should be possible to compare two Student objects. Similarly, by

```
class Student
{
private:
    string name;
    float gpa;
    string address;

public:
    Student();
    Student(string n, float g, string a);
    // copy constructor
    Student(const Student &);
    // converts the values of the instance
    // variables into a formatted string value
    string to_string() const;
    bool operator==(const Student &) const;
};

// overload the << operator for the cout object
// in order to display a Student object
ostream &operator<<(ostream &, const Student &);
```

implementing the operator << it should be possible to display a Student object by using cout.

Deliverable 8: Now, declare a DoublyLinkedList object variable, DoublyLinkedList<Student> studVector and perform the operation listed below (add screenshot demonstrating all the output of the following),

- Declare 10 Student objects and add them to the studVector variable.
- Display all the Student objects (the template based DoublyLinkedList class should display T by using cout object). Please note, Student objects should be displayed by using cout object.
- Create a copy of the studVector variable named cpStudVector (utilize the copy constructor) and display the content of the cpStudVector variable.
- Reverse the order of the Student objects in the studVector variable. Display the content of the studVector variable to ensure the reverse operation has taken place.
- Declare a Student object named stObj and demonstrate the searchItem(stObj) function. Display all the index positions where this Student object is found in the studVector variable.
- Declare a Student object named stObj and demonstrate the deleteItem(stObj) function. Display the content of the studVector variable to ensure that the delete operation has taken place.

Submit the solution on the Brightspace website by 11:59pm, Thursday, 22 Feb 2024. Let me know if you have any questions. Thank you.

Assignment rubrics Total points: 30	Points
Deliverable 1: DoublyLinkedList template class	4 points
Deliverable 2: DoublyLinkedList copy constructor	2 points
Deliverable 3: deleteItem method of the DoublyLinkedList class that removes all entries	4 points
Deliverable 4: delete(index) method + Big O	5 points
Deliverable 5: reverse the content of the DoublyLinkedList	4 points
Deliverable 6: search(searchItem) algorithm: returns all index positions of the searchItem in the DoublyLinkedList	3 points
Deliverable 7: Define and implement the Student class	4 points
Deliverable 8: Demonstrate the accuracy of the DoublyLinkedList implementation in a), b), c), d), e), f) [program + screenshots]	4 points
Miscellaneous: Javadoc comments for all the methods Syntax error in the program (program does not run) Missing screenshot	-5 points -30 points -5 points