

Issue Date: Sep 19, 2023

Due date: Sep 26, 2023

Instructions:

- Plagiarism is defined as "taking and using the thoughts, writings, and inventions of another person as one's own". IBA has no compromise policy on Plagiarism – in case, plagiarism proved student will be given **F grade**.
 - Your assignment should represent your own effort. However, you are not expected to work alone. It is fine to discuss the exercises and try to find solutions together, but each student shall write down and submit his/her solutions separately. It is good academic standard to acknowledge collaborators, so if you worked together with other students, please list their names.
 - Electronic Submission on LMS is compulsory.
 - Assignment will be graded on the basis of **timely submission on LMS**.
-

Deliverables:

Submit the JAVA code that you have written in which questions it required and submit word file for descriptive answers.

Question 1. Implementation of the basic operations of the doubly linked List

Implement a data type List that realizes linked lists consisting of nodes with integer values, as discussed in the lecture. Remember that:

- an object of type Node has three fields, an integer val, (a pointer to) a Node next and prev;
- an object of type List has one field, (a pointer to) a Node head;

The type of List must have the following methods:

1. boolean isEmpty();
2. int length(); returns the number of nodes in the list, which is 0 for the empty list;
3. void print() print the content of all nodes;
4. void addAsHead(int i) creates a new node with the integer and adds it to the beginning of the list;
5. void addAsTail(int i) creates a new node with the integer and adds it to the end of the list;
6. void addSorted(int i) creates a new node with the integer and adds it behind all nodes with a val less or equal the val of the node, possibly at the end of the list;
7. Node find(int i) returns the first node with val i;
8. void reverse() reverses the list;
9. int popHead() returns the value of the head of the list and removes the node, if the list is nonempty, otherwise returns NULL;
10. void removeFirst(int i) removes the first node with val i;
11. void removeAll(int i) removes all nodes with val i;

12. void addAll(List l) appends the list l to the last element of the current list, if the current list is nonempty, or lets the head of the current list point to the first element of l if the current list is empty.

Question 2. Modify the list with Head and Tail pointers

Implement a data type HTList that has two pointers head and tail in the lists consisting of nodes with integer values, as also discussed in the lecture. Remember that:

- an object of type Node looks as before;
- an object of type HTList has two fields, (a pointer to) a Node head and (a pointer to) a Node tail;

Modify the methods for the type List from the previous exercise (Question 2. above) so that they work for head-tail lists. You will find that some methods need not be changed at all, while for others you also have to manage the tail pointer.

Question 3. Determine the Big-Oh of each method you have implemented in both of the above questions.

	Methods	Question 2. BigOh	Question 3. BigOh
1.	boolean isEmpty()	$O(1)$	$O(1)$
2.	int length()	$O(N)$	$O(N)$
3.	void print()	$O(N)$	$O(N)$
4.	void addAsHead(int i)	$O(1)$	$O(1)$
5.	void addAsTail(int i)	$O(N)$	$O(1)$
6.	void addSorted(int i)	$O(N)$	$O(N)$
7.	Node find(int i)	$O(N)$	$O(N)$
8.	void reverse()	$O(N)$	$O(N)$
9.	int popHead()	$O(1)$	$O(1)$
10.	void removeFirst(int i)	$O(N)$	$O(N)$
11.	void removeAll(int i)	$O(N)$	$O(N)$
12.	void addAll(List l)	$O(N)$	$O(1)$