GTU DEPARTMENT OF COMPUTER ENGINEERING

CSE-222 SPRING 2023 HOMEWORK REPORT

ONUR ATASEVER

210104004087

PROBLEM SOLUTION APPROACH

In this assignment, we were asked to implement a linked list class for a basic social media software which is already designed by me.

First of all, I investigated linked list classes I saw in the lecture.

Then I created draft of class. There was a linked list class and there was also node class inside it. Each node were going to keep data just like an arraylist index.

After created draft, I decided necessary method to implement.

Because there were a class extended and it was implement an interface. I left remove and get method last because before I implement those methods, I should have created a draft for laziliy deletion in my mind. I implemented add, size, set methods and necessary get methods for Node class carefully.

I learned that what is laziliy deletion and I created a draft. According to this draft I implemented remove and get methods.

In the end I tried lots of test cases to find if there is mistakes. If there is, I fixed them.

Finally, I used LDlinkedList class in my homework.

TIME COMPLEXITY ANALYSIS

logIn

logOut

sharePost (add, setHistory)

follow(isBlocked, setFollowing, setFollowers, setHistory)

viewProfile(equals, displayPosts, set, setHistory)

likePost(equals, addInteraction, like, setHistory)

showInteractions(equals, posts.get(i).showInteractions, setHistory)

addComment(equals, addInteraction, posts.get().addComment, setHistory)

sendMessage(isBlocked, isFollowing, sendedMessage, receivedMessage, setHistory)

checkInBox

checkOutBox

viewInBox(size, .get().display)

viewOutBox(size, .get().display)

block(add, deleteFollower, deleteFollowing)

unlike(equals, removeLike, setHistory)

uncomment(removeComment, setHistory)

unfollow(isFollowing, get().equals, remove, removeFollowers, setHistory)

unblock(removeBlock , setHistory)

showHistory

	Basic Array	ArrayList	LinkedList	LDLinkedList
logIn	O(1)	O(1)	O(1)	O(1)
logOut	O(1)	O(1)	O(1)	O(1)
sharePost	O(n)	O(1)	O(n)	O(n)
follow	O(n)	O(n)	O(n^2)	O(n^2)
viewProfile	O(n)	O(n + m)	O(n^2 + m^2)	O(n^2 + m^2)
likePost	O(n)	O(n)	O(n*m)	O(n*m)
showInteraction	O(n^2)	O(n*m)	O(n*(n+m^2))	O(n*(n+m^2))
addComment	O(n)	O(n)	O(n + m)	O(n + m)
sendMessage	O(n)	O(n)	O(n + m + x)	O(n + m + x)
checkInBox	O(1)	O(1)	O(1)	O(1)
checkOutBox	O(1)	O(1)	O(1)	O(1)
viewInBox	O(n)	O(n)	O(n^2)	O(n^2)
viewOutBox	O(n)	O(n)	O(n^2)	O(n^2)
block	O(n^2)	O(n*m)	O(n+m*(m+s))	O(n+m*(m+s))
unlike	X	O(s+(n*x)+n)	O(s+p+n*(n+x))	O(s+p+n*(n+x))
uncomment	X	O(n^2)	O(s+p+n*(n+x))	O(s+p+n*(n+x))
unfollow	X	O(m*n+m)	O(n*(m+n))	O(n*(m+n))
unblock	X	O(m*n+m)	O(n(n+m))	O(n(n+m))
showHistory	X	O(n)	O(n)	O(n)
	X			

RUNNING TIME

IMPLEMENTATİON TYPE	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Basic Array Structure (HW1)	36 ms	36 ms	10 ms	
Array List Structure (a)	42 ms	40 ms	37 ms	42 ms
Linked List Structure (b)	34 ms	36 ms	37 ms	40 ms
LD Linked List Structure (c)	35 ms	38 ms	36 ms	42 ms