CSCI-UA-102-011-Spring-2025

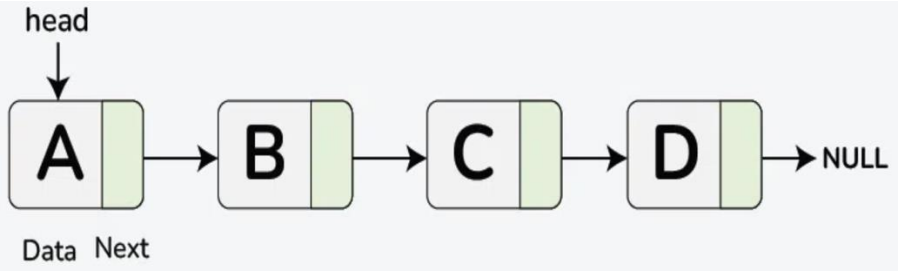
Recitation - 2

- Rasmika Billa
(01/31/2025)

Note

- Office hours: Tuesday 1:00 - 3:00PM, Location: 60FifthAve Room 204
- Slides and Materials will be uploaded in the next week (Details will be mailed)
- No Quiz Today

LinkedList



Types of Linked List:

- Singly Linked List
- Doubly Linked List
- Circular Linked List

Operations on Singly Linked List:

- Traversal
- Searching
- Length
- Insertion:
  - Insert at the beginning
  - Insert at the end
  - Insert at a specific position
- Deletion:
  - Delete from the beginning
  - Delete from the end
  - Delete a specific node

```
public class Node {
   int data;
   Node next;

   public Node(int data)
   {
      this.data = data;
      this.next = null;
   }
}
```

Node Structure

```
public static void traverseLinkedList(Node head)
{
   // Start from the head of the linked list
   Node current = head;

   // Traverse the linked list until reaching the end
   while (current != null) {

      // Print the data of the current node
      System.out.print(current.data + " ");

      // Move to the next node
      current = current.next;
   }

   System.out.println();
}
```

Traversal

Source: https://www.geeksforgeeks.org/singly-linked-list-tutorial/

Singly LinkedList

```
1   public boolean equals(Object o) {
2     if (o == null) return false;
3     if (getClass( ) != o.getClass( )) return false;
4     SinglyLinkedList other = (SinglyLinkedList) o;        // use nonparameterized type
5     if (size != other.size) return false;
6     Node walkA = head;                                    // traverse the primary list
7     Node walkB = other.head;                              // traverse the secondary list
8     while (walkA != null) {
9       if (!walkA.getElement( ).equals(walkB.getElement( ))) return false; //mismatch
10      walkA = walkA.getNext( );
11      walkB = walkB.getNext( );
12    }
13    return true;     // if we reach this, everything matched successfully
14  }
```

Code Fragment 3.19: Implementation of the SinglyLinkedList.equals method.

Check whether the current list is equal to another list based on the elements stored in the nodes and their order!

Line 1: Method Signature
Line 2: Null Check
Line 3: Class Comparison Check
Line 4: Casting Object
Line 5: Size Check
Line 6-7: Traverse Primary and secondary List
Line 8-12: Comparing Elements in both Lists

Problem Statements (15-20 mins)

- Q3.26 - Group 1
- Q3.25 - Group 2
- Q3.18 - Group 3
- Q3.17 - Group 4
- Q3.12 - Group 5
- Q3.9 - Group 6
- Q3.6 - Group 7
- Q3.5 - Group 8
- All – Q3.28

- We will start the presentation in 10 mins
- Explain your problem statement and approach

GROUPS

Page No.:
Date:

2.10

GROUP 1 → Teddy Bayan, Cathy Zhai
Ganling Zhou, Amy Kim

2.11

GROUP 2 → William Camacho, Allen Feng, Bruce Li, Jason Liu (Kaijie)

2.12

GROUP 3 → Nianchen Miao, Ian Tang, Michael Chen, Brian Shen

2.13

GROUP 4 → Hewitt Ho, Ryan Yamamoto, Xan Carey, Colin Lee

2.14

GROUP 5 → Ali Guo, Ryan Lu, Justin Gao, Haelyn Ryoo

2.21

GROUP 6 → Christopher Cajamarca, Zhuoran Wen
Alexander Flores, Gavin Zhou

2.24

GROUP 7 → Serena Kher, Meili Liang
Jacqueline Wu, Aashain Singhal

2.29

GROUP 8 → ~~Jacqueline Wu~~
Victor Kao
Omer Mosker
Jeffrey Solano
Grace Yin
David Wu