UNIVERSITY OF CALOOCAN CITY
Caloocan, 1400 Metro Manila, Philippines

COLLEGE OF ENGINEERING
Computer Engineering
1st Semester, School Year 2025-2026

Data Structure & Algorithm

Assignment No. 1

Singly Linked List

Submitted by:

Villanueva, Bryan O.

7:00AM – 10:00AM / BSCPE 2-A

Submitted to

Engr. Maria Rizette H. Sayo

Instructor

Date Performed:

08/11/2025

Date Submitted

08/11/2025

I. Objectives

In this section, the goals in this laboratory are:

- To define a variant of a linked list
- To be able to know the real-world application of it

II. Methods

1. What is a singly linked list, and how does it differ from an array?

   A singly linked list is a linear data structure made up of nodes where each node contains two fields, data and a reference or pointer to the next node in the sequence. This structure contrasts with an array, which stores elements in contiguous memory locations. While arrays provide direct access to elements via indices, singly linked lists require sequential traversal from the head node to reach a specific element. The flexibility of linked lists allows dynamic memory allocation, making insertion and deletion operations more efficient compared to arrays, which may require shifting elements.

2. When would you prefer a linked list over an array, vice versa?

   I will prefer using a linked list over an array if my program needs to do a lot of insertions and deletions, especially not just at the end, or if I don't know exactly how many elements I'll have and it needs to change often. Linked lists can handle these changes more efficiently without having to move a lot of data around. But if I need quick access to elements by their index or I'm working with a fixed number of items, I'd go with an array since it's faster for that and easier to manage in memory.

3. How are linked lists used in real-world applications (e.g., browser history, undo functionality)?

   Linked lists find numerous practical applications in real world scenarios. For example, browsers often use linked lists to manage their history data, where each visited webpage is stored as a node, allowing users to navigate backward and forward easily. Similarly, undo functionality in text editors frequently employs linked lists to keep track of changes, enabling users to revert or reapply actions. Additionally, linked lists are used in dynamic memory allocation, managing free memory blocks, and implementing other data structures like stacks and queues.

III. Conclusion

In conclusion, I understand that linked lists and arrays are both useful, but they have different strengths. Linked lists are better when I need flexibility for adding or removing data, while arrays are better for fast access to fixed data. I also learned that linked lists are not just for theory they are used in real life, like in browser history, undo features, and memory management. Knowing when and how to use each one can help me write better and more efficient programs.

Reference

Website

GeeksforGeeks. (2013, March 8). *Basic Terminologies of Linked List*. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/what-is-linked-list/