**Data Structure & Algorithm**

Assignment No. 1

**Singly Linked List**

*Submitted by:*

**Eulin, Ryan Bertrand B.**

**7:00AM – 10:00 / BSCPE 2-A**

*Submitted to*

**Engr. Maria Rizette H. Sayo**

Instructor

*Date Performed:*

**15/08/2025**

*Date Submitted*

**15/18/2025**

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?

Singly linked list is a linear data structure made up of nodes where in a node stores data (the values we want) and a pointer or the reference point for the next node. Singly linked list differs from an array in a way that, array is a collection of elements stored in a contiguous memory location meaning it is placed side by side, while singly linked list is a variant of linked list where in the main point is that the data is linked to each other. So, they’re not side by side, they can be scattered in the memory.

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

If you need constant-time insertions/deletions from the list (such as in real-time computing where time predictability is absolutely critical) use linked list. Else you know the number of elements in the array ahead of time so that you can allocate the correct amount of memory for the array use array

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

Many applications, like text editors, use singly linked lists to implement the "Undo" feature. Each action is stored as a node in the list, and when you undo, the program navigates back through each node sequentially. This approach keeps memory usage efficient and ensures that actions are easily reversible (Regan, 2024).

III. Conclusion

To sum it all together, I’ve learned that singly linked lists differ to an array, how I can apply it when we’re building an app and its advantage to array and vice versa. Singly linked list has its own strength and weaknesses differs to the problem where facing and in all the data structure we may encounter has their own role their build to a different problem therefore, we can use it to plan on what data structure we will to the system where making.

**Reference**

Website

[1] Blue Tree Code. (2019, May 1). *Singly Linked List | Insert, Delete, Complexity analysis* [Video]. YouTube. <https://www.youtube.com/watch?v=Ovhj6qDSF9M>

[2GeeksforGeeks. (2025, July 23). *Linked List vs Array*. GeeksforGeeks. <https://www.geeksforgeeks.org/dsa/linked-list-vs-array/>

[3] Neso Academy. (2020, June 24). *Array vs. Single Linked List (In Terms of Representation)*

[Video]. YouTube. <https://www.youtube.com/watch?v=b5QR4AmrspU>

[4] *Regan, I. (2024, November 4). Singly Linked Lists : Understanding with Real-Life Examples.* [*https://www.linkedin.com/pulse/singly-linked-lists-understanding-real-life-examples-infant-regan-m7vnc#:~:text=Many%20applications%2C%20like%20text%20editors,that%20actions%20are%20easily%20reversible*](https://www.linkedin.com/pulse/singly-linked-lists-understanding-real-life-examples-infant-regan-m7vnc#:~:text=Many%20applications%2C%20like%20text%20editors,that%20actions%20are%20easily%20reversible)*.*

[5] *When to use a linked list over an array/array list?* (n.d.). Stack Overflow. <https://stackoverflow.com/questions/393556/when-to-use-a-linked-list-over-an-array-array-list/>