UNIVERSITY OF CALOOCAN CITY
Caloocan, 1400 Metro Manila, Philippines

COLLEGE OF ENGINEERING
Computer Engineering
2nd Semester, School Year 2024-2025

Singly Linked List and Array

Laboratory Activity No. 1

Review of Singly Linked List and Array

Submitted by:
Palmes, Lewis Clark L.
Saturday / BS CpE 1-A

Submitted to
Engr. Maria Rizzete Sayo
Instructor

Date Performed:
16-08-2025

Date Submitted
16-08-2025

I. Objectives

In this section, the goals in this laboratory are:

- *To define the singly linked list*
- *To be able to know how singly linked list used in real world application*

II. Methods

General Instruction: Discuss the following questions provided below:

1. What is a singly linked list, and how does it differ from an array?
2. When would you prefer a linked list over an array, and vice versa?
3. How are linked lists used in real-world applications (e.g., browser history, undo functionality)?

4. Cite your reference/s

III. Results

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

A singly linked list is a linear data structure where each element, called a node, contains two parts:

- ➢ Data - the actual value stored.

- ➢ Pointer (next) - a reference to the next node in the sequence.

The last node points to NULL, indicating the end of the list.

Difference from an array:

- ➢ Memory storage: Arrays store elements in contiguous memory locations; linked lists store elements in scattered memory connected by pointers.

- ➢ Size flexibility: Arrays have fixed size (must be declared or resized), while linked lists can dynamically grow or shrink.

- ➢ Access speed: Arrays allow direct access to elements via index (O(1)), while linked lists require traversal from the head node (O(n)).

- ➢ Insertion/Deletion: Easier and faster in linked lists (O(1) if position is known) compared to arrays, which may require shifting elements (O(n)).

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

Prefer linked list when:

- ➢ Frequent insertion or deletion in the middle or beginning of the sequence.
- ➢ Memory allocation needs to be dynamic.
- ➢ The total number of elements is not known in advance.

Prefer array when:

- ➢ Fast random access to elements is needed.
- ➢ The size is known and fixed.
- ➢ Memory overhead must be minimized (arrays don't store extra pointer fields).

3. How are linked lists used in real-world applications?

- ➢ Browser history: Each visited webpage is a node; moving forward/back uses linked list traversal.
- ➢ Undo functionality in text editors: Each edit is stored as a node, and undo steps back through the list of changes.
- ➢ Music playlists: Songs stored as nodes; you can go to the next or previous song.
- ➢ Memory management: Free blocks of memory tracked using linked lists in operating systems.
- ➢ Chaining in hash tables: Linked lists handle collisions by storing multiple values in the same hash bucket.

IV. Conclusion

Singly linked lists and arrays are both essential data structures, each with unique strengths and trade-offs. While arrays offer fast, direct access and efficient storage for fixed-size data, singly linked lists provide flexibility through dynamic memory allocation and efficient insertion or deletion of elements. In real-world applications, singly linked lists are often used where data changes frequently or when memory usage needs to be managed dynamically, such as in browser history tracking, undo operations, and memory management systems. Understanding how and when to use each structure is crucial for building efficient, adaptable programs and solving computational problems effectively.

Reference

Algorithms + data structures=programs : Wirth, Niklaus : Free Download, Borrow, and Streaming : Internet Archive. (1976). Internet Archive. https://archive.org/details/algorithmsdatast0000wirt

GeeksforGeeks. (2025, July 23). Singly linked list tutorial. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/singly-linked-list-tutorial/

Introduction To Algorithms Third Edition (2009) : by Thomas H Cormen (Author), Charles E Leiserson (Author), Ronald L Rivest (Author), Clifford Stein (Author) : Free Download, Borrow, and Streaming : Internet Archive. (2024, January 11). Internet Archive. https://archive.org/details/introduction-to-algorithms-third-edition-2009