Adoracion, Jerick Dave D. Engr. Maria Rizette Sayo

BSCpE 2B

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

According to Geeksforgeeks.com, a singly linked list is a special type of linked list in which each node has only one link that points that points to the next linked list. Each node has two parts:

* **Data** stores the actual value.
* **Next** – a reference (or pointer) to the next node in the list.

The list starts at a **head** node and ends at a node whose *next* pointer is *null* (or *None* in Python).

An **array** stores elements in a contiguous block of memory, allowing **direct (random) access** to any element using its index, but requiring costly **shifting operations** for insertions or deletions in the middle. In contrast, a **singly linked list** stores elements as separate **nodes** scattered across memory, where each node contains data and a pointer to the next node. This structure enables **efficient insertions and deletions** once the position is found, without shifting elements, but accessing an element requires **sequential traversal** from the head of the list, making it slower for direct lookups. Arrays are generally more memory-efficient per element, while linked lists trade extra pointer storage for flexibility in dynamic resizing.

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

In terms of linked list, one of the best situation I can think of is whenever I’m constructing a playlist on spotify, whether to delete or insert my favorite songs, it’s best to use linked list because I don’t have to shift all the other songs from memory. Just update a couple of pointers, and you’re done. This makes linked lists ideal when your program will frequently insert or remove elements in the middle of a collection.

While in terms of arrays, it’s best if we’re tracking a record. For example, we’re tracking a daily attendance of the BSCpE 2B class. I can use Array in order to jump at a specific date as the data are stored contiguously, so it’s memory efficient. I get **instant access** to any reading by its index without looping through the whole list.

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

Some of the examples I can think of are, whenever you are scrolling through your gallery, you are using linked list function without ever noticing it. When viewing photos, each image is linked to the next and previous images in a sequence. This allows easy navigation without reloading or searching through the entire collection.

Also, just like in what I’ve said in number 3, whenever you are constructing a playlist, for example, in Spotify, you are using a linked list function while inserting and deleting a certain song. You can easily add, remove, or reorder songs without shifting all other items like you would in an array.

***Citations:***

**GeeksforGeeks**, “Singly Linked List Tutorial,” GeeksforGeeks, 23-Jul-2025. [Online]. Available: https://www.geeksforgeeks.org/dsa/singly-linked-list-tutorial/. [Accessed: 15-Aug-2025].

**GeeksforGeeks**, Linked List vs Array – GeeksforGeeks, GeeksforGeeks, 3-weeks ago. [Online]. Available: https://www.geeksforgeeks.org/dsa/linked-list-vs-array/. [Accessed: 15-Aug-2025]

**Sahil Mattoo**, Difference Between Array and Linked List: Key Comparisons, IntelliPaat, 30-Jul-2025. [Online]. Available: https://intellipaat.com/blog/difference-between-array-and-linked-list/. [Accessed: 15-Aug-2025]

Linked list, **Wikipedia**, 3 days ago. [Online]. Available: https://en.wikipedia.org/wiki/Linked\_list. [Accessed: 16-Aug-2025]

*type of in which e*

*e has only one link that points to the next node in the linked list.*