



Linked Lists vs. Arrays

Let's put the two data structures against each other to find out which is more efficient.

We'll cover the following



- Array vs. Linked List
 - Memory Allocation
 - Insertion and Deletion
 - Searching

Array vs. Linked List

Memory Allocation

The main difference between a linked list and an array is the way they are allocated memory. Arrays instantiate a whole block of memory, e.g., `array[1000]` gets space to store 1000 elements at the start even if it doesn't contain any element yet. On the other hand, a linked list only instantiates the portion of memory it uses.

Insertion and Deletion

For lists and arrays, many differences can be observed in the way elements are inserted and deleted. In a linked list, insertion and deletion at head happen in a constant amount of time ($O(1)$), while arrays take $O(n)$ time to insert or delete a value because you have to shift the array elements left or right after that operation.

Searching

In an array, it takes constant time to access an index. In a linked list, you have to iterate the list from the start until you find the node with the correct value.

The table given below will summarize the performance difference between linked lists and arrays.

Operation	Linked List	Array
Access	$O(n)$	$O(1)$
Insert (at head)	$O(1)$	$O(n)$
Delete (at head)	$O(1)$	$O(n)$
Insert (at tail)	$O(n)$	$O(n)$
Delete (at tail)	$O(n)$	$O(n)$

As you can see, there is a trade-off between the facilities provided by both structures. You will understand more about the working of *linked lists* in the lessons that follow.

In the next lesson, let's learn about another commonly used variation of a linked list called the doubly linked List.

[← Back](#)[Next →](#)[Solution Review: Deletion in Singly Lin...](#)[What is a Doubly Linked List \(DLL\)?](#)☒ **Mark as Completed**[Report an Issue](#)[Ask a Question](#)https://discuss.educative.io/tag/linked-lists-vs-arrays__linked-lists__data-structures-for-coding-interviews-in-java