Universidade Federal de Ouro Preto
Lecture Notes
Arrays, Linked Lists, Staks, Queues

Prof. Rodrigo Silva

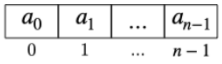
April 3, 2023

Source

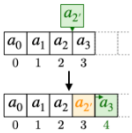
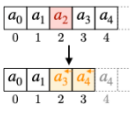
- Arrays `https://superstudy.guide/algorithms-data-structures/data-structures/arrays-strings`
- Stacks and Queues `https://superstudy.guide/algorithms-data-structures/data-structures/stacks-queues`

1 Arrays

An array is a collection of elements of the same data type that are stored together in contiguous memory locations and can be accessed using an index or a subscript.

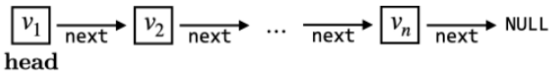


1.1 Array operations

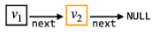
Type	Time	Description	Illustration
Access	O(1)	Using index i, we can directly access the ith element of the array with A[i].	
Search	O(n)	We need to search the array by checking each element one by one until finding the desired value.	
Insertion	O(n)	1. Elements at indices i and up are moved to the right. 2. The new element is inserted at index i. *Note that if there is no space for the new element to be added in the existing array, we need to create a bigger array and copy existing elements over there.*	
Deletion	O(n)	1. Elements at indices i+1 and up are moved to the left. 2. The former last element of the array is either ignored or removed.	

2 Linked List (Lista Encadeada pt-BR)

A linked list is a linear data structure that consists of a sequence of nodes, each containing a value or data and a pointer to the next node in the sequence. The last node typically has a null pointer, indicating the end of the list.

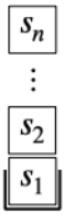


2.1 Linked List operations

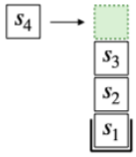
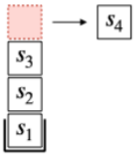
Type	Time	Description	Illustration
Access	O(n)	Starting from the head, the linked list is traversed in a sequential way to access the desired node. If it is the last element, the whole linked list is traversed.	
Search	O(n)	Starting from the head, the linked list is traversed in a sequential way to search for the desired node. If the node is not found, the whole linked list is traversed.	
Insertion	O(1)	1. Assign the next pointer of the previous-to-be element to the newly-inserted node. 2. Assign the next pointer of the new element to the following element.	
Deletion	O(1)	Assign the next pointer of the previous node to the following node of the soon-to-be-removed node.	

3 Stacks (Pilha pt-BR)

A stack is an abstract data type that represents a collection of elements with a particular set of operations. It is based on the principle of Last-In-First-Out (LIFO), which means that the last element added to the stack is the first one to be removed.

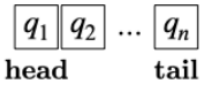


3.1 Stack operations

Push	Pop
Insert an element on the top of the stack.	Remove the element from the top of the stack and return its value.
	

4 Queues (Fila pt-BR)

A queue is an abstract data type that represents a collection of elements with a particular set of operations. It is based on the principle of First-In-First-Out (FIFO), which means that the first element added to the queue is the first one to be removed.



4.1 Queue operations

Enqueue	Dequeue
Insert element at the tail of the queue.	Remove element from the head of the queue.