# Data structures and algorithms

## Arrays

* Always the same length in memory.
* A continuous block stored in memory.
* All its elements are the same size.
* To get an element of an array knowing the index it is always the same time complexity since the formula is just the starting memory block plus the index of the element times the element size, meaning the time complexity would be O(1).
* If you don’t know the index of the element and you want to find a specific element you would have to iterate through the array to find it, so the time complexity would be O(n).
* Adding an element to a full array would be O(n) because you would need to create a new array, copy the original elements and adding the new one.
* Adding an element to the end of an array (with space) is O(1) since we have the index.
* Inserting, updating or deleting an element at a specific index is also O(1).
* Deleting an element by shifting elements or if you don’t have the index you would have O(n).

**Bubble Sort**

* We start with the unsortedPartitionIndex in the last index of the array
* And we have another index (i) in the start of the array. If the element in i is greater then the one on the right we swap the element, otherwise we leave them, and increment i+1.
* At the end of the iteration i = unsortedPartitionIndex.
* Then we pace the unsortedPartitionIndex in unsortedPartitionIndex – 1 since the last element is already ordered and i = 0. And we start all over again. We do this until unsortedPartitionIndex = 0.
* It’s an inplace algorithm, although we create extra variables, it does not depend on the number of elements we are sorting so it does not use any extra memory.
* It has an O( time complexity, so in terms of time it is not very effective.