Introduction

In this lesson, we'll go over some basic array concepts, most of which you might already know.

Arrays

Arrays are an integral part of many advanced algorithms and data structures. Many of them will require you to know how to perform operations, how to perform optimizations, and what the limitations are.

We'll go over some commonly used terms and their meaning.

Subarray

A subarray is a contiguous part of the array. I'll be using this notation to denote subarray $A[i...j]$. This represents a subarray of the array A containing all the elements from ith to jth index, both inclusive. For example:

$$A = [1, 2, 3, 4]$$

$$A[1..3] = [2, 3, 4]$$

An array of size N has an order of N^2 subarray. $\frac{N*(N-1)}{2} + N$ to be exact.

N subarrays of size 1 and $\binom{N}{2}$ subarrays of size > 1 (Pick any 2 index)

Subsequence

A subsequence of an array can be obtained by deleting some (or none, or all) elements of the original array without changing the order of elements. For example:

Few subsequences for $A = [1, 2, 3, 4]$ are $[1, 2, 3]$, $[1]$ including an empty array $[]$.

$[3, 1]$ is not a subsequence because the order is not the same as the original array.

An array of size N has 2^N subsequences, meaning for each element in the array, we have 2 choices (keep it or remove it).

Contiguous memory allocation

Array elements occupy contiguous memory blocks, and we have the following properties as a result:

Element access is $O(1)$ operation

Since we know the starting address, any arbitrary element's address is simple arithmetic.

Inserting element in between is $O(N)$ operation

Consider you have an array of size 10 allocated and it only has 5 elements $A[0...4]$. To insert an element at position 2, we have to shift $A[2..4]$ to the right by one place. In the worst case, we'll have to shift the entire array (*N operations*).

In the next lesson, we'll see what vectors are and how to use them.