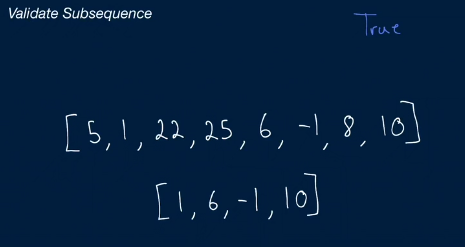


Subsequence concept is very important in interviews.

A subsequence is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements.



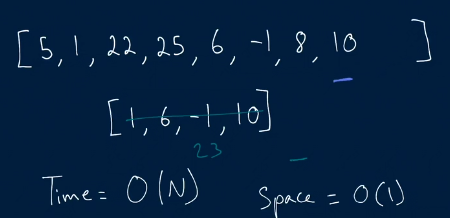
How to solve this?

First we have to realize that we have to traverse the both arrays that were given.

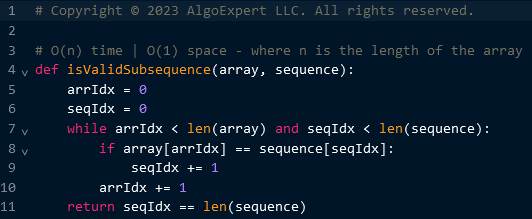
How do we traverse them?

Subsequence cares about the order of elements.

* So we’re looking for the 1st element which is 1 here.
* So we’re gonna initialize a pointer underneath the 1st element of our subsequence.
  + Now, let’s traverse our main array until we find this first element that our pointer is pointing to.
  + After finding the 1st element of subsequence in main array, we will continue to look all other elements.
  + After finding the 1st element of subsequence in main array, increase the pointer index for both array iterations by 1.



Solution 1:

* Use a while loop to traverse through the both array in tandem and is gonna keep track of the positions that we are at in both arrays.
* As we have to track the positions in both arrays, initialise array index and sequence index variables to 0.
* Inside the while loop:
  + We’re looking to see if the element that we are at in the main array is equal to the element that we are at in the subsequence array.
    - If equals then move the position in the subsequence by 1. (increment sequence index by 1)
  + Regardless of the above condition, increment array index by 1.
  + If sequence index reaches the length of sequence then we can say subsequence is present in main array.
* O(n) time
* O(1) space (here we are storing only 2 variables for indices and nothing that exceeds the size of input arrays.)
* 

Solution 2:

* Use for loop to traverse through the main array and is gonna keep track of our position only in the 2nd array.
* 