

Arrays & Strings

Arrays

An array is a number (finite set) of data items of the same type arranged contiguously in memory, it is the most commonly used data storage structure. Array's elements can be ordered or not. An array must meet the following conditions:

- **Must be finite:**

It means that an array must have a certain size, so its data is determined. To use an array, its size must be determined by the user, this is a disadvantage of an array whereas if a certain size of memory area is reserved for an array, and this reserved area isn't used completely, it can not be used for another variable, and it stays empty.

- **Fixed locations:**

The locations of an array stay fixed and not changed during dealment with any item of an array.

- **Must be sequential:**

The locations of an array must be sequential.

- **Randomly accessed:**

It can be randomly access to any location of an array, without depending on any location, so the required time for accessing any location is fixed.

- **Data must be homogenous:**

The data can not be in different types in the same array, it must be with the same data type like integer, string, ..., etc.

Some Applications on Array

Ex. 1: Insertion an element to sorted (ordered) array.

Insertion an element x to sorted array $A[0..n-1]$, where all the elements of the array are sorted in increasing order before and after insertion process.

Algorithm *Insertelement* (A, n, x):

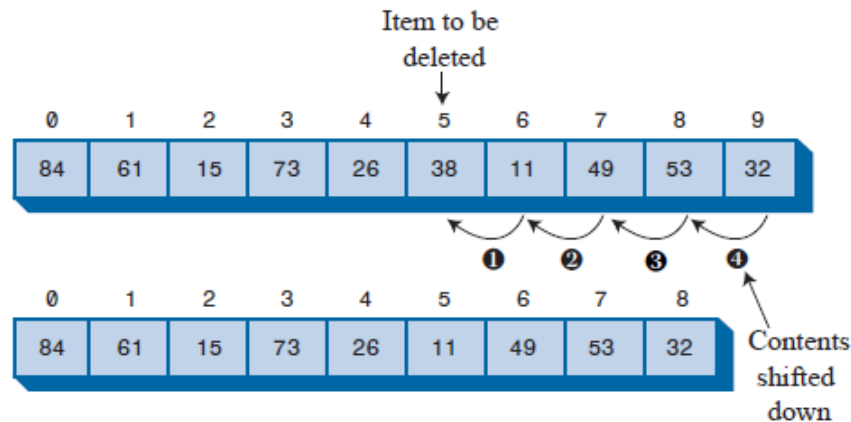
Input: Sorted array $A[0..n-1]$ with size n and the required element x to be inserted.

Output: Sorted array $A[0..n]$.

1. for $i \leftarrow n-1$ down to 0 while $i \geq 0$ and $x < A[i]$
2. $A[i+1] \leftarrow A[i]$ // shifting up (to right)
3. end for
4. $A[i+1] \leftarrow x$ // insert
5. $n \leftarrow n+1$

Ex. 2: Delete an element from array (sorted or not).

To delete an element from an array (it is ordered or not), you need to look for an element, and you must shift down all the elements that lie after an item to be deleted, as below:



Note:

The size of the array does not change, but in dealing with the array after deleting an element, you must ignore the excess (فائض) locations, but in fact, they exist and reserved.

* **H.W.** // Write the algorithm of deleting an item from an array.

Ex. 3: Sequential Searching for an Item

To search for an item, scan the array with comparing the item we're looking for with each element. If an array is scanned with no match being found, this means that the value isn't in the array, then display a message that indicates that.

Sequential search is one of search types, it searches about x element in array A .

Algorithm SeqSearch(A, x):
Input: An array $A[0..n]$ such that the n elements are stored in positions from 1 to n , and an element x .
Output: An index i such that $x=A[i]$, or the special value(0) if x does not appear in A .

1. $i \leftarrow n$
2. $A[0] \leftarrow x$
3. **while** $A[i] \neq x$
4. $i \leftarrow i-1$
5. **end while**
6. **return** i

Strings

String is a set of symbols, such as the following examples :

ABCDEF

1234567


+ - * / =

THIS IS A STRING

&5{ABCDEH!

Operations on strings

The main applied operations on strings are:

- 1- **Assignment**: Assign string in another string, such as $str1=$  $str2$.
- 2- **Concatenation**: Merge the content of two string to get a new string, such as merge (concatenate) the string "abc" with string "def", the new string is "abcdef".

- 3- **Pattern Matching:** Search about appearance a string in another string, such as the string "Base" appears in string "Modern Base" from the position 8.
- 4- **Substring:** Get a substring by determining the two points of dividing (i.e. the first position of dividing and the last position) in the original string.
- 5- **Insertion:** Insert a symbol or substring in the original string, by determining the insert position in the original string. Such as insert the string "Second" in the string "This is Lecture" in the position 9 to get a new string "This is Second Lecture".
- 6- **Deletion:** Determining a symbol or substring that is required for deletion it from the original string, by knowing the first position of the substring.

Storing The Strings

The string can be store using one of the following methods based on the used data structure.

- 1- Fixed Length String (using arrays).
- 2- Linked Lists.
- 3- Index Tables.

In the **first method**, a fixed number of memory locations is reserved, and the symbols of a string are stored in those locations. In the case of existing excess (فائض) locations, they are reserved and stay empty, and can not be used for another data. Such empty locations is one of the disadvantages of this storing method, because it leads to a waste in memory and does not exploit it in a perfect form. This method uses array data structure for storing any data type such as string (set of symbols), integers, doubles, float (real numbers), etc.

The **second and third** method will be covered in the subsequent lecture to store any data type.

Some Applications on Strings (represented with array)

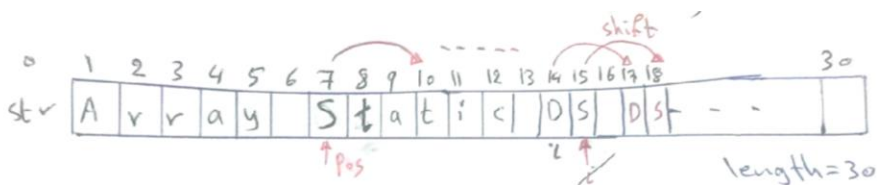
Ex. 1: Insertion

To insert a string in another string, follow the steps:

- The length of the added string with the length of the original string must not exceed the original string's length.
- Determine the start position of insertion (pos).
- Shift the original string's elements to the right, to provide a space for the string's elements that be inserted.

For example:

To insert string "is" in the string "Array Static DS" in position 7



* $15 + 3 = 18 \leq 30$ Do insert

* $i = 15$

* shift $\left\{ \begin{array}{l} \text{str}(i+3) = \text{str}(i) \Rightarrow \text{str}(18) = \text{str}(15) \\ i = i - 1 = 14 \end{array} \right.$

while $i \geq \text{pos}$

$\text{str}(14+3) = \text{str}(14)$

$\text{str}(7+3) = \text{str}(7)$ ($i = \text{pos} = 7$ أول)

$i = 7 - 1 = 6$ توقف التكرار

Now insert "is" from $\text{pos} = 7$

$\text{str}(\text{pos}) = \text{str}(7) = i$ $\text{pos}++$

$\text{str}(8) = s$ $\text{pos}++$

$\text{str}(9) = \text{ } (space)$

New string is

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
A r r a y i s t a t i c D S

Ex. 2: Concatenation

Merge two strings in a new string as the following steps:

- Check the summation of the length of two strings with the length of the third string that will contain the result of concatenation, if it is enough do the next steps.
- Store the elements of the first string.
- Store the elements of the second string.

