

Algoritmos y Estructura de Datos I

Unidad 2

Ejercitación: Operaciones elementales sobre arreglos

Parte I

A partir de la estructura **Array** definida en el módulo **algo1**, crear un módulo con el nombre **array.py** que **implemente** las siguientes especificaciones de las operaciones elementales para el **TAD secuencia** utilizando arreglos.

- **search(Array,element)**

Descripción: Busca un elemento en el Array que representa el **TAD secuencia**,

Entrada: el Arreglo sobre el cual se quiere realizar la búsqueda (Array) y el elemento (element) a buscar.

Salida: Devuelve el índice donde se encuentra la primera instancia del elemento. Devuelve **None** si el elemento no se encuentra dentro del array.

- **insert(Array,element,position)**

Descripción: Inserta un elemento en una posición determinada de un Array que representa el **TAD secuencia**.

Poscondición: Se desplazan todos los demás elementos hacia el final. El elemento en la última posición del Array se pierde.

Entrada: el arreglo (Array) sobre el cual se va a hacer la inserción, el elemento (element) y la posición (position) donde se quiere insertar.

Salida: Si pudo insertar con éxito devuelve la posición donde se inserta el elemento. En caso contrario devuelve **None**. Devuelve **None** si la posición a insertar es mayor que el número de elementos en el array.

- **delete(Array,element)**

Descripción: Elimina un elemento del arreglo que representa el **TAD secuencia**.

Poscondición: Se desplazan los elementos restantes y se rellena con **None** hacia el final.

Entrada: el arreglo sobre el cual se quiere realizar la eliminación (Array) y el elemento (element) a eliminar.

Salida: Devuelve la posición donde se encuentra el elemento a eliminar. Devuelve **None** si el elemento a eliminar no se encuentra.

- **length(Array)**

Descripción: Calcula el número de elementos **activos** que hay en la **secuencia**

Entrada: El arreglo sobre el cual se quiere calcular el número de elementos

Salida: El número de elementos distintos a None

Parte II

Verificar la funcionalidad del módulo desarrollado utilizando el archivo **array_unittest.py**. Para realizar el test usando **repl.it** el archivo se deben crear 2 archivos. El archivo **main.py** debe contener todo el código de test de **array_unittest.py** y luego abrir otro archivo con nuestro módulo **array.py** que contendrá todas las operaciones de sobre arreglos.

Resolución:

myarray.py × +

```
1 #####
2 #funciones o subprocessos
3 #####
4 import algo1
5
6 def search(Array,element):
7     n=len(Array)
8     for i in range(0,n):
9         if Array[i]==element:
10             return i
11     return None
12
13 def insert(Array,element,position):
14     n=len(Array)
15     if position>=n:
16         return None
17     ArrayR=algo1.Array(n,0)
18     for i in range(0,n):
19         ArrayR[i]=Array[i]
20     for i in range(position,n-1):
21         ArrayR[i+1]=Array[i]
22     ArrayR[position]=element
23     for i in range(0,n):
24         Array[i]=ArrayR[i]
25     return position
26
27 def delete(Array,element):
28     position=search(Array,element)
29     if position==None:
30         return None
31     n=len(Array)
32     ArrayR=algo1.Array(n,0)
33     for i in range(0,position):
34         ArrayR[i]=Array[i]
35     for i in range(position,n-1):
36         ArrayR[i]=Array[i+1]
37     for i in range(0,n):
38         Array[i]=ArrayR[i]
39     return position
40
41 def length(Array):
42     cont=0
43     n=len(Array)
44     for i in range(0,n):
45         if Array[i]!=None:
46             cont+=1
47     return cont
48
```

[Link Repl.it \(con unittest\):](https://replit.com/@Paulonia/Martinez13866U2Array#myarray.py)

<https://replit.com/@Paulonia/Martinez13866U2Array#myarray.py>