



Análisis de Complejidad Teórica

Todos los ejercicios son obligatorios

- 1) Calcular la cantidad de OE (operaciones elementales) para cada una de las operaciones del TAD secuencia implementado sobre arreglos:

Access(Array,posicion)

Search(Array,Element)

Insert(Array,element,posicion)

Delete(Array,element)

- 2) Calcular el orden de complejidad $O(f)$ para cada una de las operaciones del ejercicio 1.

Access -> $O(1)$

Search -> $O(n)$

Insert -> $O(n)$

Delete -> $O(n)$

- 3) Calcular el orden de complejidad $O(f)$ para los siguientes códigos:

Codigo 1 $O(1)$

1.	if a>b:
2.	c=a+b
3.	else:
4.	for d in range(1,10):
5.	c=a+b*d

Codigo 2 $O(n)$

1.	a=1
2.	while a<n:
3.	a=a+1

Codigo 3 $O(n^2)$

1.	for i in range(1,n):
2.	j=0
3.	while j<i:
4.	a=a*(1+j)
5.	j=j+1



Codigo 4 $O(n^3)$

1.	for a in range (1,n):
2.	for b in range (a,n):
3.	if L[a]==L[b]:
4.	delete(L,L[b])

1)

Access(Array,posición) -> 6OE

```
def access(array, position): # 30E
    return array[position] # 30E
```

Search(Array,Element) -> 10 OE + 7n OE

```
def search(array, element): # 30E

    encontrado = False # 1 OE
    for i in range(0, len(array)): # 1 OE + 5n OE + n(S1)
        if array[i] == element: # S1 = 10E + 10E
            return i # 20E

    if encontrado == False: # 10E
        return None # 20E

    #T(search) = 30E + 10E + 10E + 5nOE + n(20E) + 20E + 30E = 10 + 7n OE
```

Insert(Array,element,posición) -> 22 OE + 7n OE

```
def insert(array, element, index): #40E

    if out_of_range(index, 0, len(array)): #70E + 50E
        return None #20E

    else:

        for i in range(len(array)-1, index, -1): #40E + 3n
            array[i] = array[i-1] # 4nOE

        array[index] = element # 20E

    #T(insert) = 40E + 120E + max{20E, 60E + 7n} = 160E + 60E + 7n = 220E + 7nOE
```

Delete(Array,element) -> 15 OE + 17n OE

```
def delete(array,element): # 3

    encontrado = False # 1

    for i in range(0,len(array)): # 1 + 5n
        if array[i] == element: # 2n
            encontrado = True # 1n
            index = i # 1n

    if encontrado: # 1
        for i in range(index,len(array)-2): # 1 + 1 + 4n
            array[i] = array[i+1] # 4n

        array[len(array)-1] = None # 5
        return index # 2
    else:
        return None

#T(delete) = 15 + 17n
```