

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

Access(Array,posicion)

```

1 def acces(Array, position): 3OE
2     return Array[position] 3OE

```

TOTAL: 6OE

Search(Array,Element)

```

3 def search(Array, element): 3:3OE
4
5     index=0                    5:1OE
6     for i in Array:           6:2nOE
7         if i==element:       7:1OE
8             return index     8:2OE
9         index=index+1        9:2OE
10
11     return None              11:1OE

```

TOTAL: 10+2n OE

Insert(Array,element,posicion)

```

13 def insert(Array, element, position): 13:4OE
14
15     if position<0 or position>=len(Array): 15:4OE
16         return None 16:1OE
17     else:
18         for i in range (len(Array)-1,position, -1): 18:2+n
19             Array[i]=Array[i-1] 19:n
20
21         Array[position]=element 21:1OE
22         return position 22:2OE

```

TOTAL: 13+2n OE

Delete(Array,element)

```

25 def delete(Array, element): 25:3OE
26
27     index=search(Array, element) 27:4OE
28     if index != None: 28:1OE
29         for i in range(index, len(Array)): 29:2+nOE
30             if i<len(Array)-1: 30:3n
31                 Array[i]=Array[i+1] 31:3n
32             elif i==len(Array)-1: 32:3n
33                 Array[i]=None 33:2OE
34         return index 34:2OE
35     else:
36         return None 36:1OE

```

TOTAL: 15+10n

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

Access(Array, posicion) = $O(1)$

Search(Array, Element) = $O(n)$

Insert(Array, element, posicion) = $O(n)$

Delete(Array, element) = $O(n)$

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

Codigo 1

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

$O(1)$

Codigo 2

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

$O(n)$

Codigo 3

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

$O(n^2)$

Codigo 4

```
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])
```

$O(n^3)$