

5.20

buscarCoincidencia(lista): boolean

 inicio=0

 fin=lista.largo -1

 medio=(inicio+fin)//2

 si (lista.get(medio) == medio)

 devolver verdadero

 sino

 si (lista.get(medio)>medio)

 fin=medio-1

 sino

 inicio=medio+1

 FinSi

 FinSi

Devolver falso

Fin

Su tiempo de ejecución es de $O(\log n)$, ya que se va reduciendo a la mitad la cantidad de elementos con cada bucle.

5.23

Buscarmayoritario(lista)

Posible=null

Contador=0

Para cada elemento e en lista

Si (contador==0)

Posible=e

Contador=1

Sino

Si (e==posible)

Contador++

Sino

Contador - -

FinSi

FinSi

FinPara

Ocurrencias=0

Para cada elemento e en lista

Si (e==posible)

Ocurrencias++

FinSi

FinPara

Si ((ocurrencias>lista.largo)/2)

Devolver posible

Sino

Devolver -1

FinSi

Fin

El algoritmo recorre la lista dos veces, una para encontrar un candidato a mayoritario y otra para contar cuántas veces aparece, por lo tanto su tiempo de ejecución es lineal, es decir, **$O(n)$** , donde n es la cantidad de elementos en la lista.