

TEMAS

Intercambio de valores de una lista

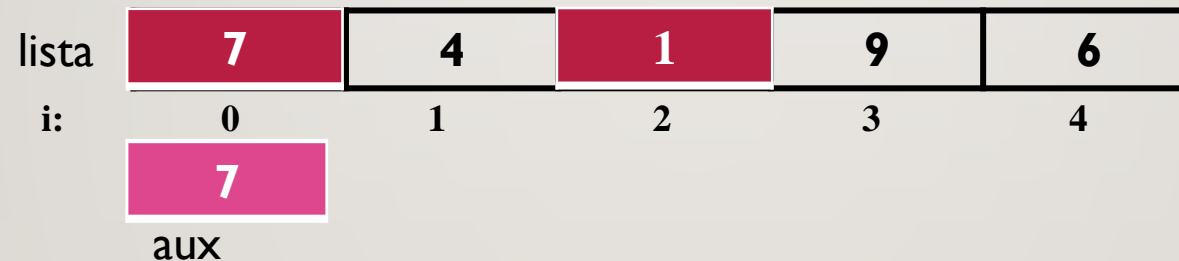
Métodos de Ordenamiento: Selección

Búsqueda Binaria

INTERCAMBIO

PARA ORDENAR UN VECTOR ES NECESARIO REALIZAR INTERCAMBIOS DE ELEMENTOS

Para intercambiar dos valores en Pos1 y Pos2 ,
se necesita una variable auxiliar.



Paso 1: copiar en aux el valor de `vec[Pos1]`

`aux = lista[Pos1]`

Paso 2: copiar en `vec[Pos1]` el valor de `vec[Pos2]`

`lista[Pos1] = lista[Pos2]`

Paso 3: copiar en `vec[Pos2]` el valor de aux

`lista[Pos2] = aux`

EL VALOR DE Pos1 y Pos2 DEPENDE DE LOS ELEMENTOS QUE SE QUIEREN INTERCAMBIAR

MÉTODOS DE ORDENAMIENTO



BURBUJEO

SELECCION



INSERCIÓN



SELECCION

CONSISTE EN

BUSCAR EL MENOR ELEMENTO DE TODA LA LISTA Y LO INTERCAMBIA CON EL DE LA PRIMER POSICION.

LUEGO SE BUSCA EL SEGUNDO ELEMENTO MENOR Y SE LO INTERCAMBIA CON EL DE LA SEGUNDA POSICION Y ASI SUCESIVAMENTE



SELECCION

4	9	1	6	7
---	---	---	---	---

i:

0

1

2

3

4

COMENZANDO DESDE Posición CERO, BUSCA EL MENOR

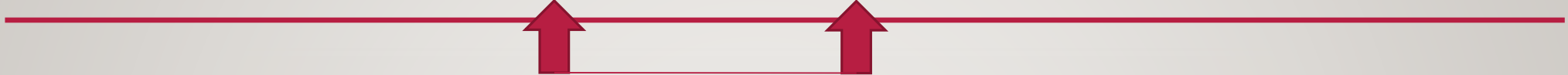
Compara la posición CERO con la posición UNO: Intercambia



SELECCION

1	9	4	6	7
0	1	2	3	4

i:



COMENZANDO DESDE Posición CERO, BUSCA EL MENOR

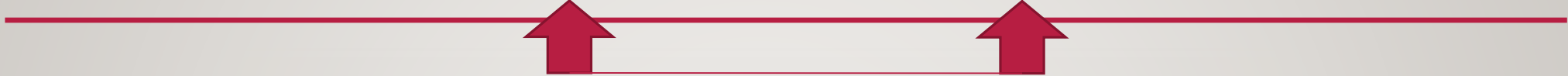
Compara la posición CERO con la posición UNO: Intercambia

Compara la posición CERO con la posición DOS: Intercambia



SELECCION

1	9	4	6	7
i: 0	1	2	3	4



COMENZANDO DESDE Posición CERO, BUSCA EL MENOR

Compara la posición CERO con la posición UNO: Intercambia

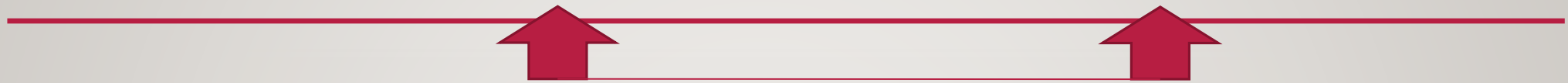
Compara la posición CERO con la posición DOS: Intercambia

Compara la posición CERO con la posición TRES: NO Intercambia



SELECCION

1	9	4	6	7
i: 0	1	2	3	4



COMENZANDO DESDE Posición CERO, BUSCA EL MENOR

Compara la posición CERO con la posición UNO: Intercambia

Compara la posición CERO con la posición DOS: Intercambia

Compara la posición CERO con la posición TRES: NO Intercambia

Compara la posición CERO con la posición CUATRO: NO Intercambia



SELECCION

1	4	9	6	7
---	---	---	---	---

i:

0

1

2

3

4

COMENZANDO DESDE EL SEGUNDO ELEMENTO, BUSCA EL MENOR

Compara la posición UNO con la posición DOS: NO Intercambia

MÉTODOS DE ORDENAMIENTO



SELECCION

1	4	9	6	7
---	---	---	---	---

i:

0

1

2

3

4

COMENZANDO DESDE EL SEGUNDO ELEMENTO, BUSCA EL MENOR

Compara la posición UNO con la posición DOS: NO Intercambia

Compara la posición UNO con la posición TRES: NO Intercambia

MÉTODOS DE ORDENAMIENTO



SELECCION

1	4	9	6	7
---	---	---	---	---

i:

0

1

2

3

4

COMENZANDO DESDE EL SEGUNDO ELEMENTO, BUSCA EL MENOR

Compara la posición UNO con la posición DOS: NO Intercambia

Compara la posición UNO con la posición TRES: NO Intercambia

Compara la posición UNO con la posición CUATRO: NO Intercambia



SELECCION

1	4	6	9	7
---	---	---	---	---

i:

0

1

2

3

4

COMENZANDO DESDE EL TERCER ELEMENTO, BUSCA EL MENOR

Compara la posición DOS con la posición TRES: Intercambia

MÉTODOS DE ORDENAMIENTO



SELECCION

1	4	6	9	7
---	---	---	---	---

i:

0

1

2

3

4

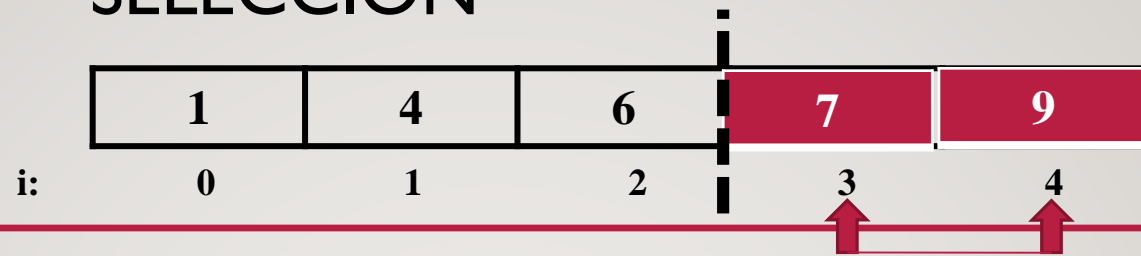
COMENZANDO DESDE EL TERCER ELEMENTO, BUSCA EL MENOR

Compara la posición DOS con la posición TRES: NO Intercambia

MÉTODOS DE ORDENAMIENTO



SELECCION



COMENZANDO DESDE EL CUARTO ELEMENTO, BUSCA EL MENOR

Compara la posición TRES con la posición CUATRO: Intercambia



SELECCION

1	4	6	7	9
0	1	2	3	4

i:

QUEDARON TODOS LOS ELEMENTOS ORDENADOS.

Se recorre la lista, se toma el primer elemento y se compara contra el resto. Si el elemento es mayor, se intercambia de lugar. De esta manera, al finalizar el proceso, el menor quedará colocado en el lugar que le corresponde.



SELECCION

1	4	6	7	9
0	1	2	3	4

i:

QUEDARON TODOS LOS ELEMENTOS ORDENADOS.

```
def metododeseleccion(v):  
    largo = len(v)  
    for i in range(largo - 1):  
        for j in range(i+1, largo):  
            if v[i] > v[j]:  
                aux = v[i]  
                v[i] = v[j]  
                v[j] = aux
```

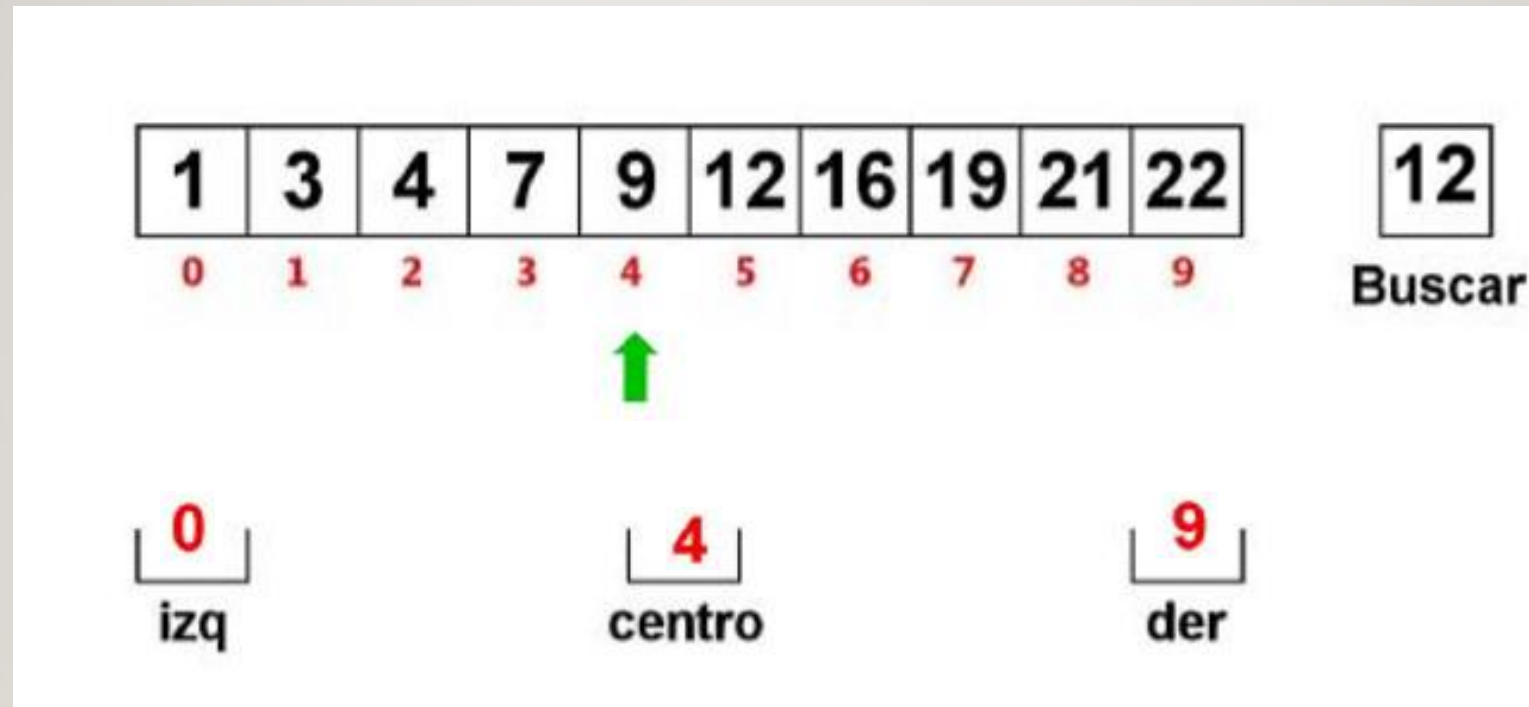

BÚSQUEDA BINARIA

LA LISTA DEBE ESTAR ORDENADA PARA USAR BÚSQUEDA BINARIA.

Procedimiento:

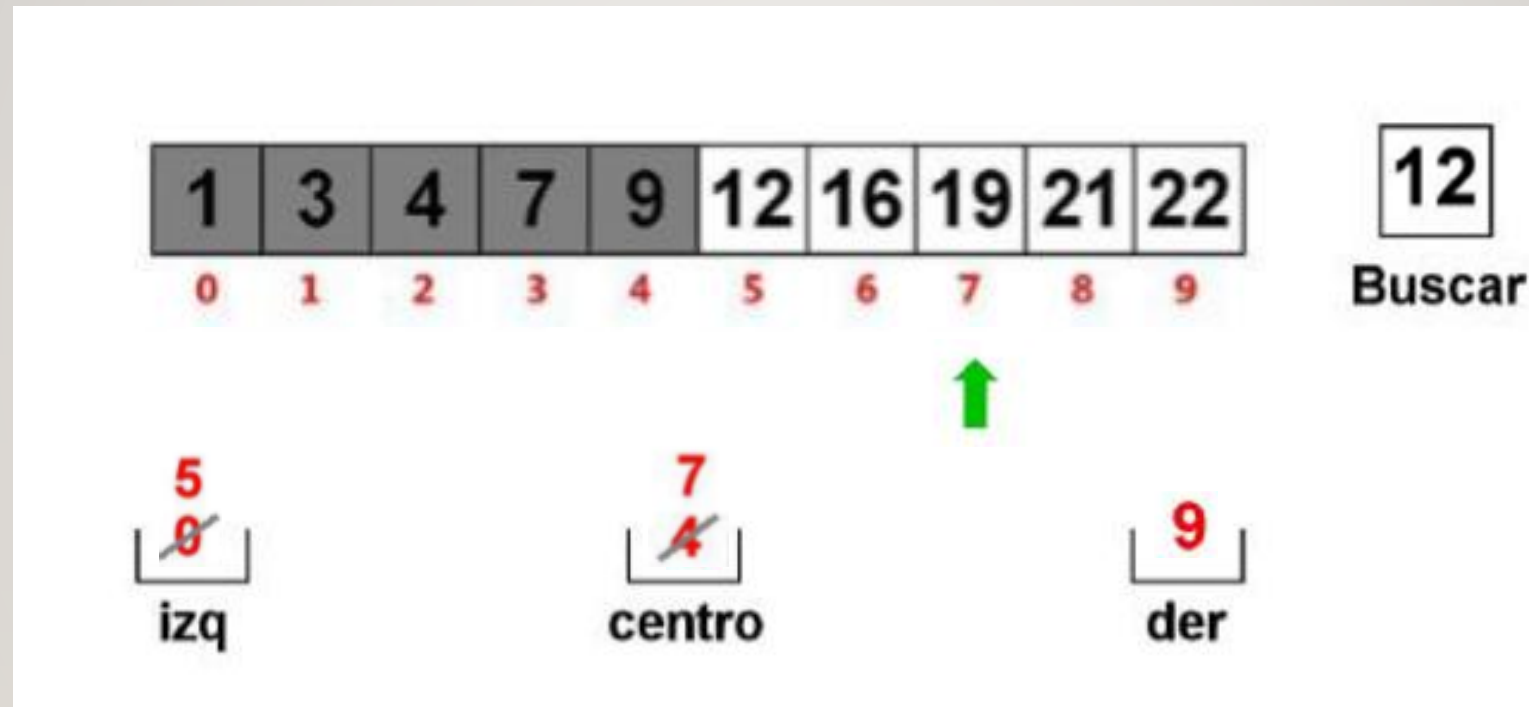
- ✓ Se verifica si en la mitad de la lista se encuentra el elemento buscado.
- ✓ Si no está, resulta fácil deducir para que lado podría estar por el ordenamiento.
- ✓ Se descarta una mitad y se repite el proceso sobre la otra mitad.

BÚSQUEDA BINARIA



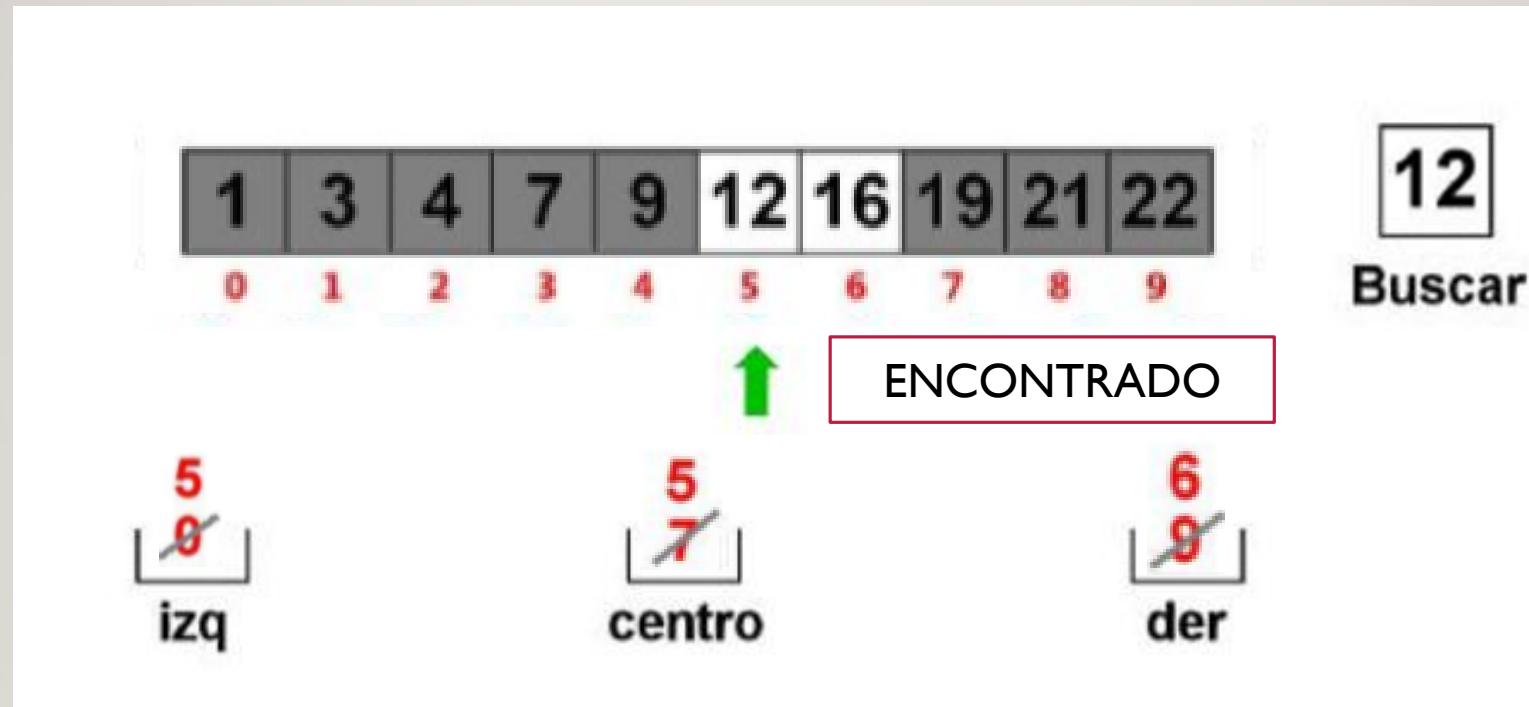
$$\text{centro} = (\text{izq} + \text{der}) // 2$$

BÚSQUEDA BINARIA



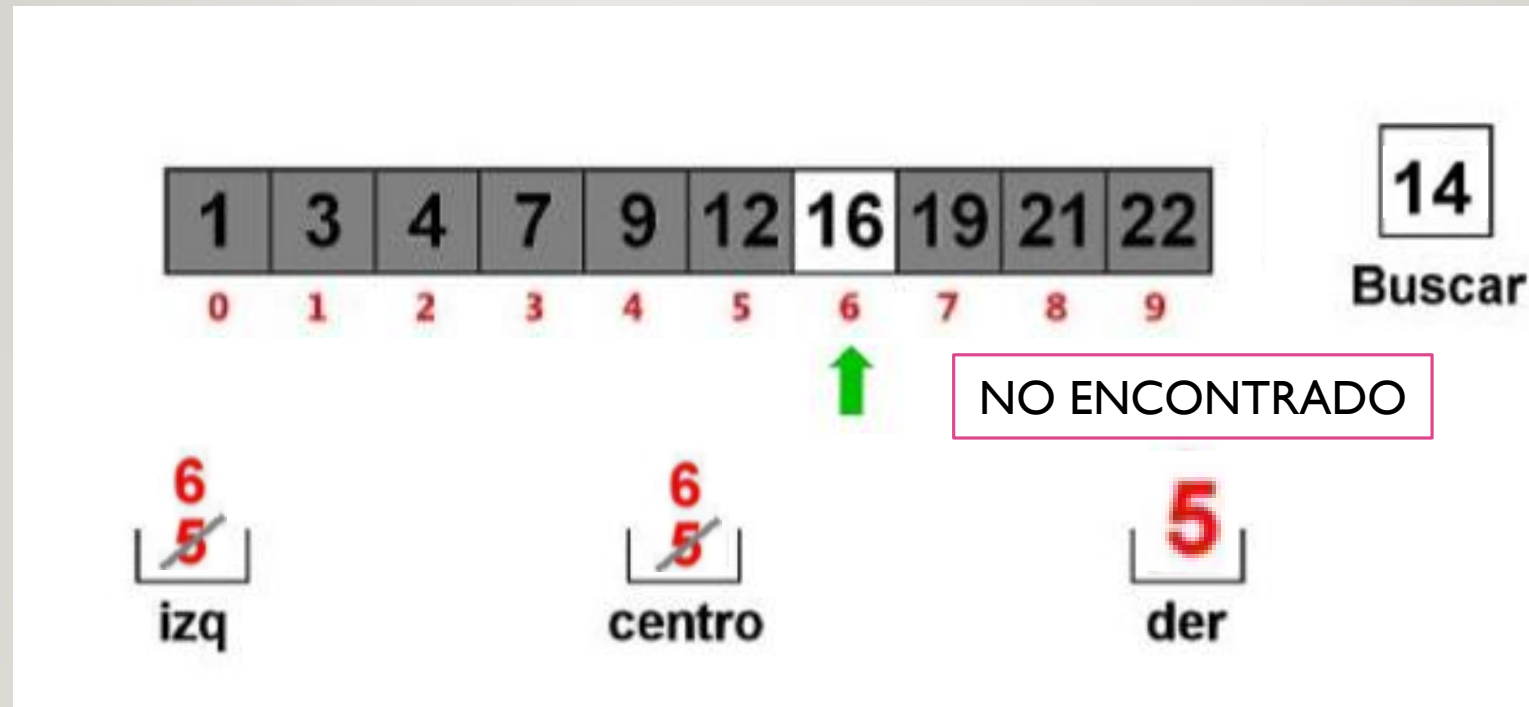
$$\text{centro} = (\text{izq} + \text{der}) // 2$$

BÚSQUEDA BINARIA



$$\text{centro} = (\text{izq} + \text{der}) // 2$$

BÚSQUEDA BINARIA



$$\text{centro} = (\text{izq} + \text{der}) // 2$$

BÚSQUEDA BINARIA

```
def busquedabinaria(v, dato):  
    izq = 0  
    der = len(v) - 1  
    pos = -1  
    while izq <= der and pos == -1:  
        centro = (izq + der) // 2  
        if v[centro] == dato:  
            pos = centro  
        elif v[centro] < dato:  
            izq = centro + 1  
        else:  
            der = centro - 1  
    return pos
```



PRACTICA!

Ingresar los legajos y las notas de final de los alumnos de un curso, Ingresar primero la cantidad de alumnos. Se solicita:

- a) Validar el ingreso de las notas, considerando una nota válida entre 1 y 10.
- b) Al finalizar el ingreso de notas, emitir un listado informando legajo y nota, ordenado por legajo de mayor a menor.
- c) Informar La Máxima nota y los legajos de todos los que se sacaron la mayor nota.
- d) Ingresar un legajo y si existe, informar su nota. Utilizar Búsqueda Binaria.

Crear las funciones que se consideren necesario para resolver el problema.