



SEC. POSGRADO Y EDUCACIÓN CONTINUA

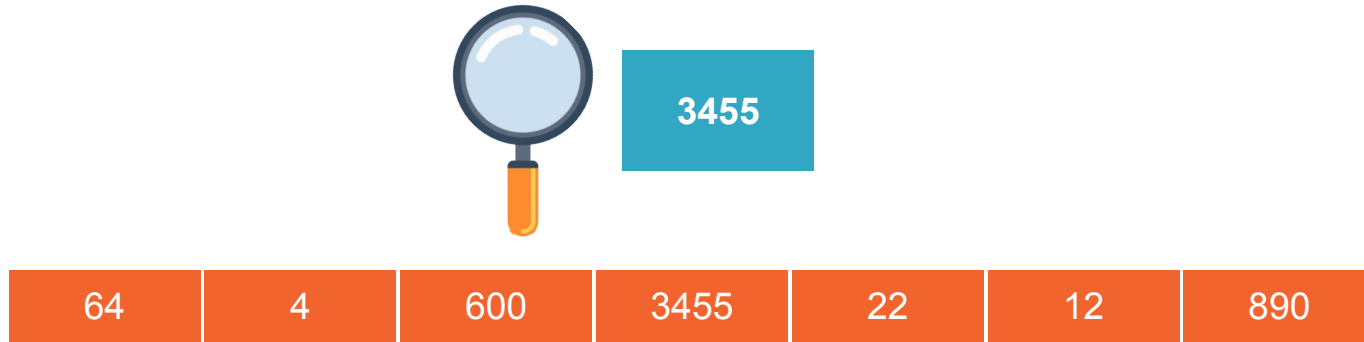
# Métodos de búsqueda

algoritmos de búsqueda

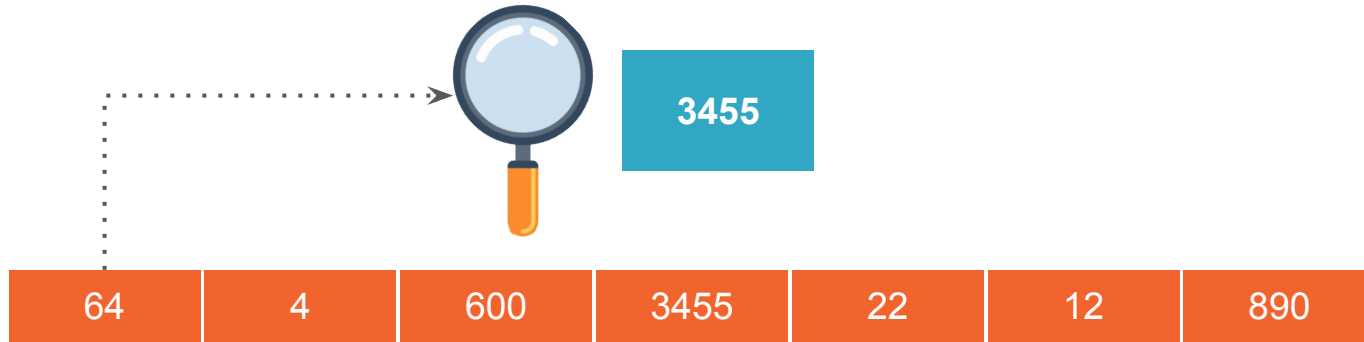
- Búsqueda lineal
- Búsqueda binaria



# Búsqueda lineal

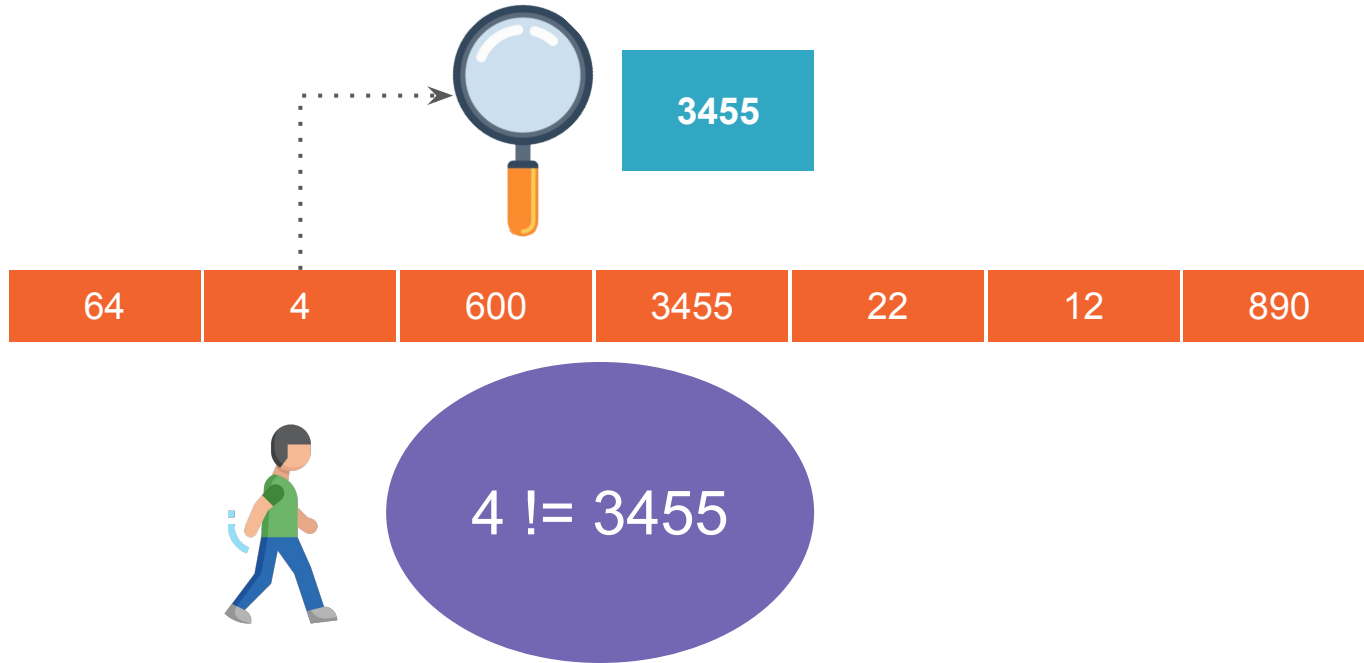


# Búsqueda lineal

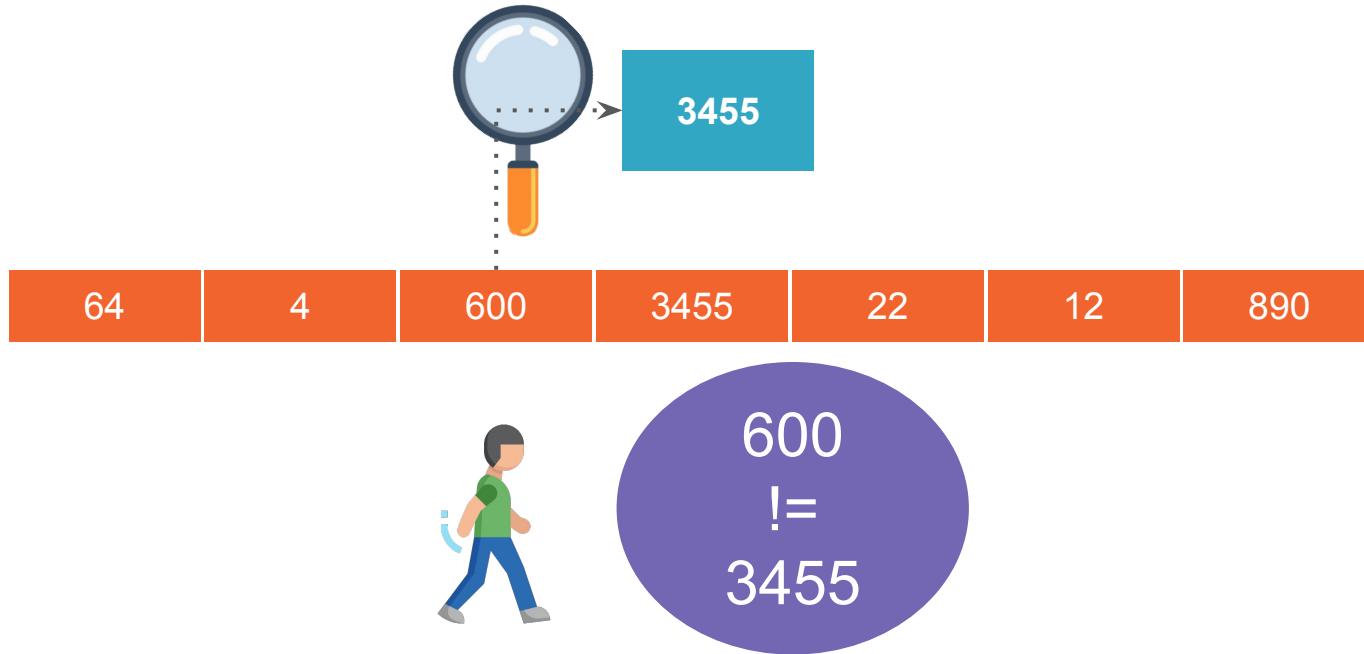


$64 \neq 3455$

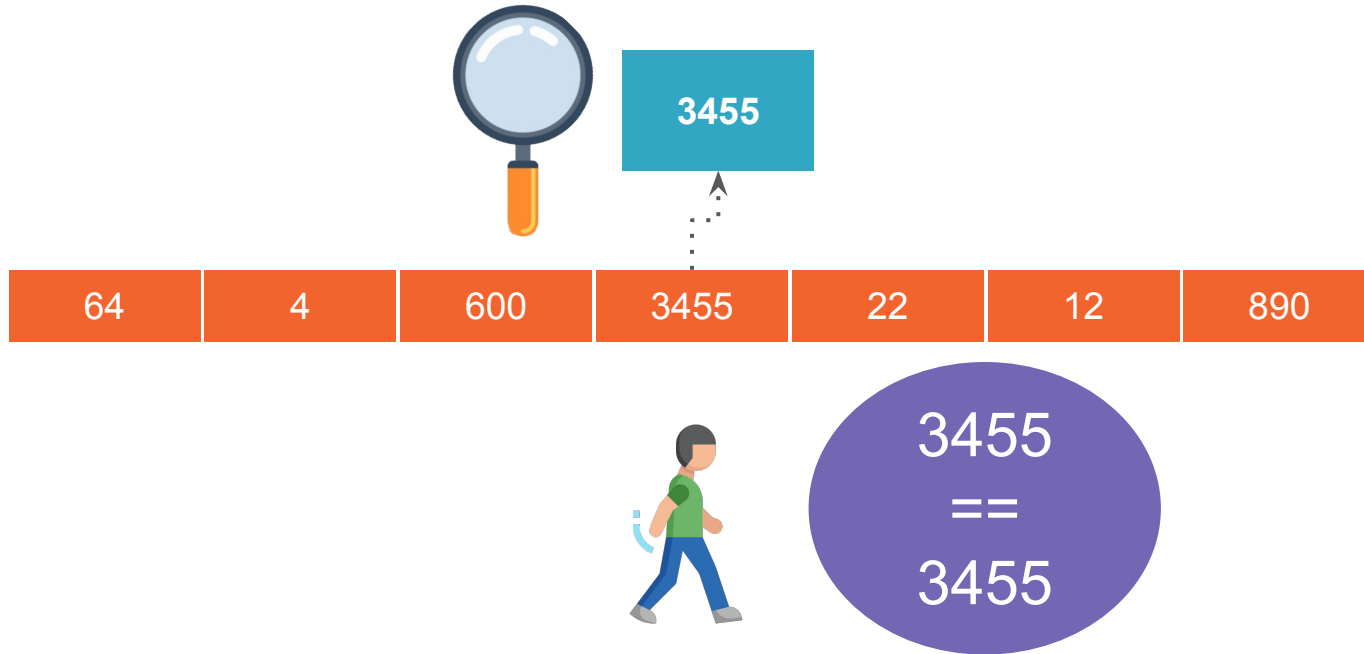
# Búsqueda lineal



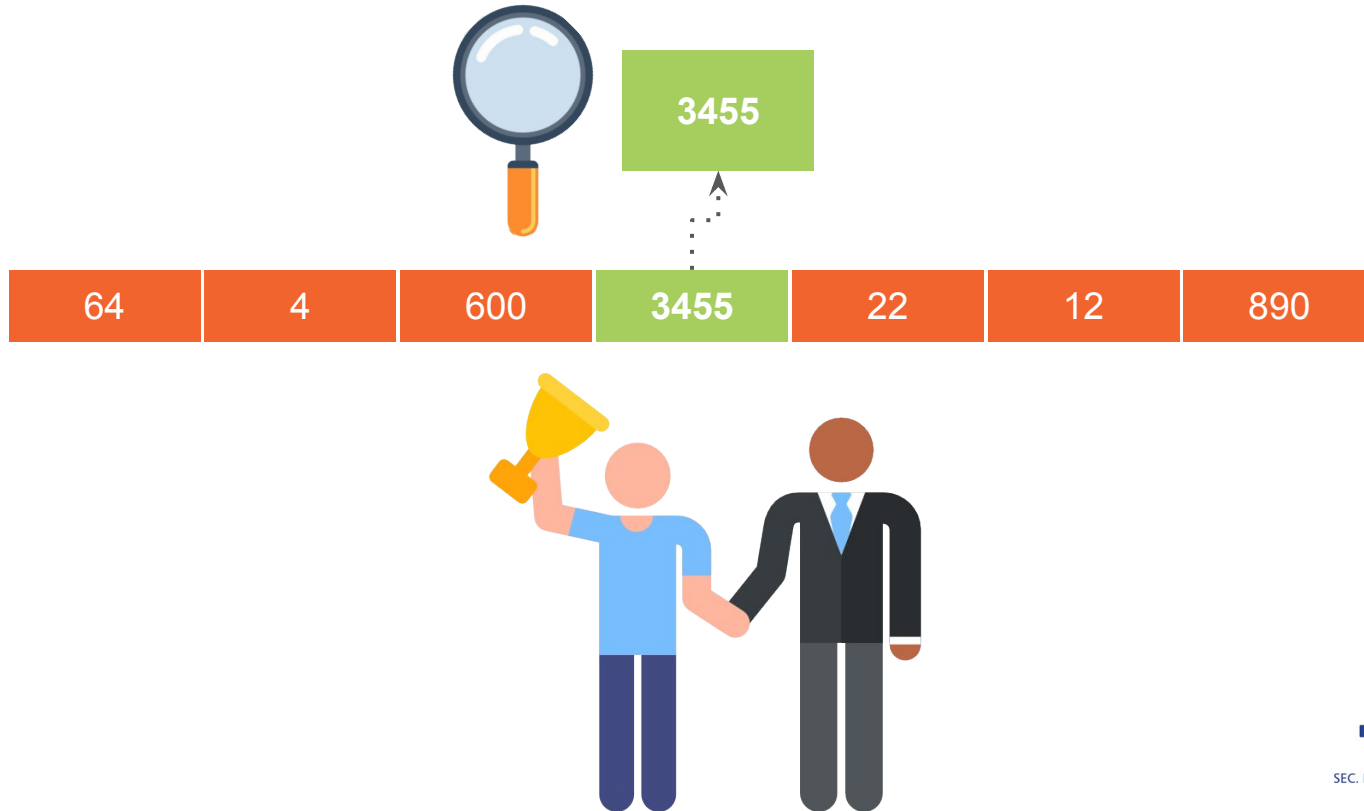
# Búsqueda lineal



# Búsqueda lineal



# Búsqueda lineal

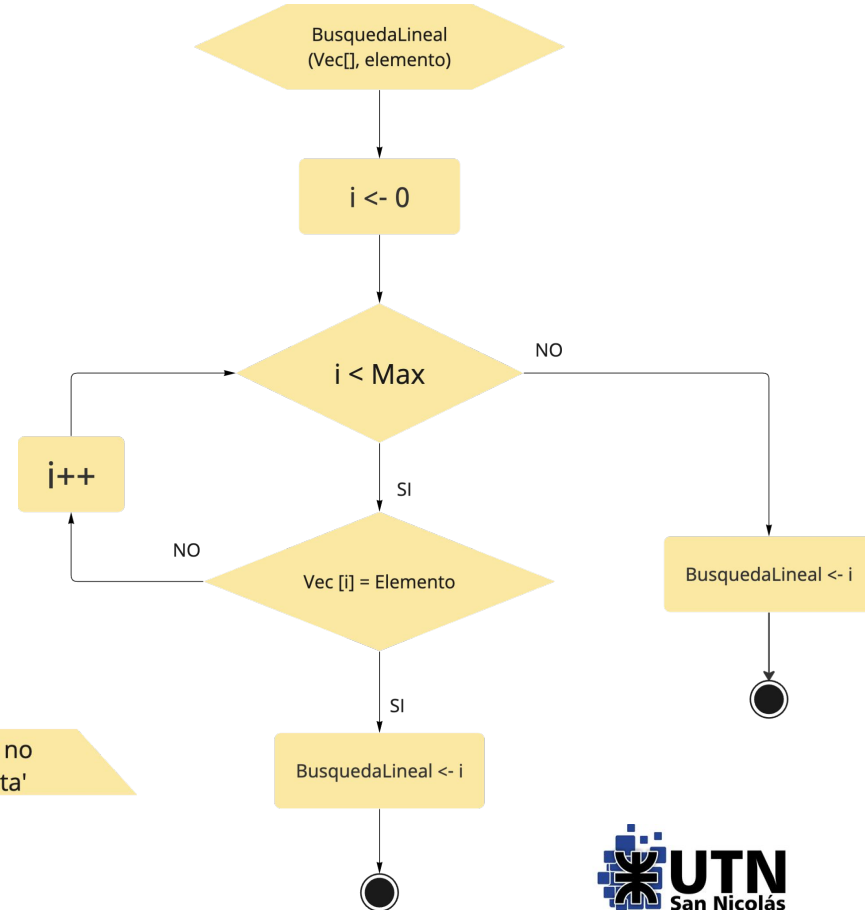
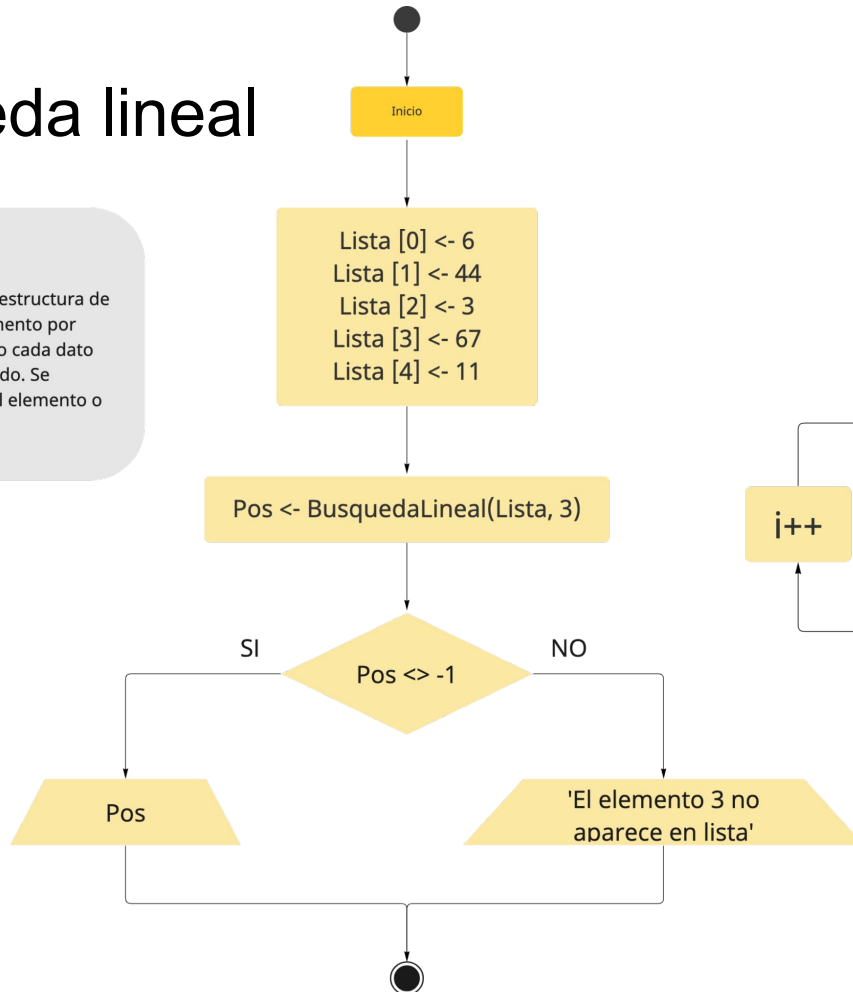




# Búsqueda lineal

## Búsqueda lineal

Se basa en recorrer la estructura de forma secuencial, elemento por elemento, comparando cada dato con el elemento buscado. Se termina al encontrar el elemento o recorrer toda la lista

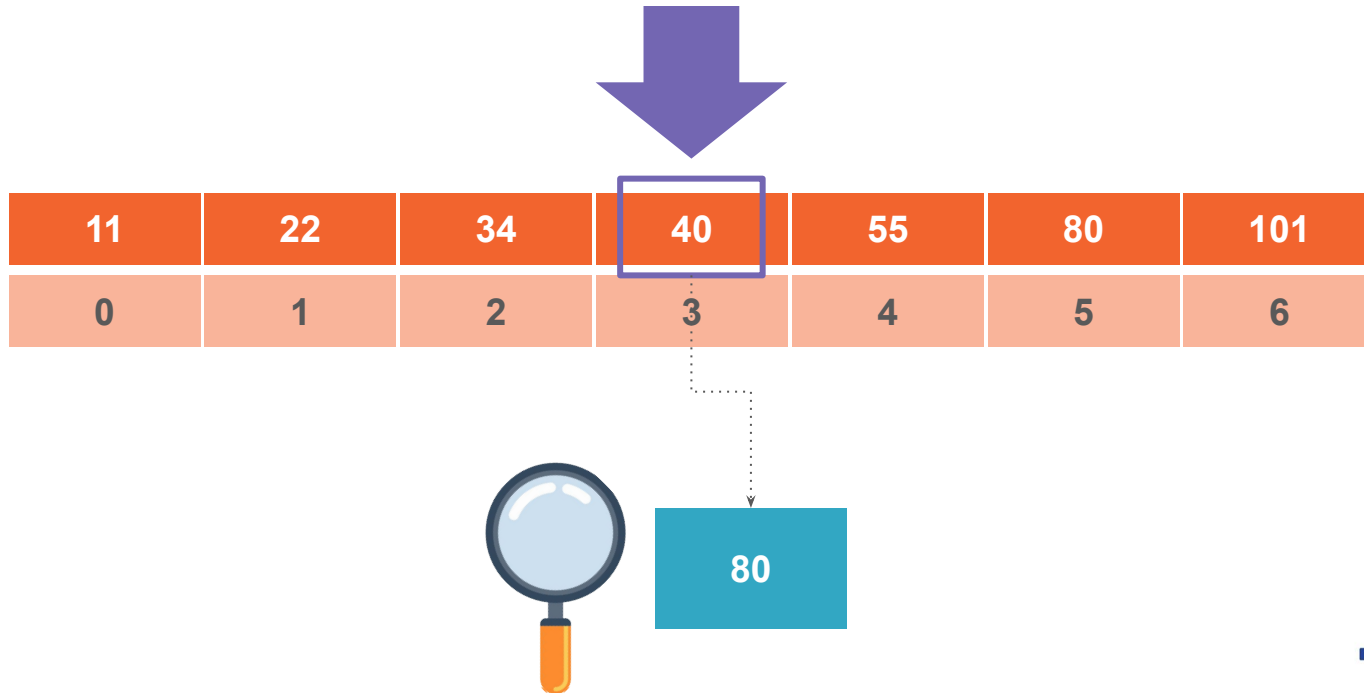


# Búsqueda binaria


Elementos ordenados

11	22	34	40	55	80	101
0	1	2	3	4	5	6

# Búsqueda binaria



# Búsqueda binaria



11	22	34	40	55	80	101
0	1	2	3	4	5	6

ítem buscado (80)

>

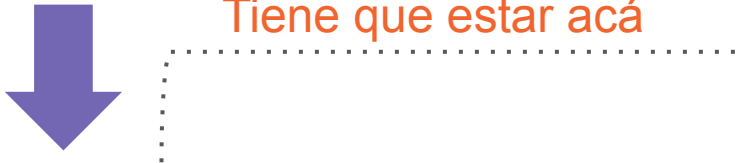
ítem analizado (40)



80

# Búsqueda binaria

Tiene que estar acá



11	22	34	40	55	80	101
0	1	2	3	4	5	6

ítem buscado (80)

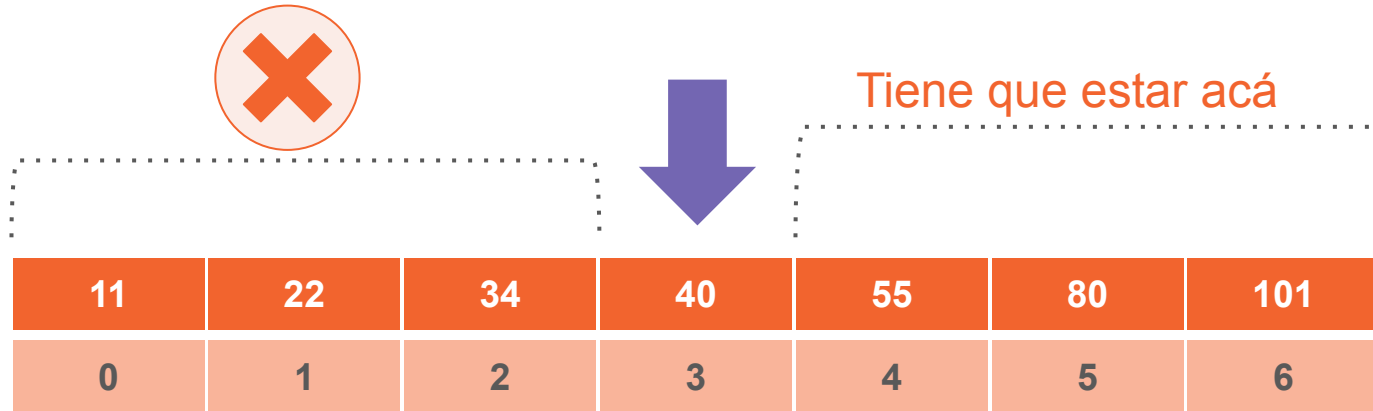
>

ítem analizado (40)



80

# Búsqueda binaria



ítem buscado (80)

>

ítem analizado (40)



80

# Búsqueda binaria

11	22	34	40	55	80	101
0	1	2	3	4	5	6



80

# Búsqueda binaria

11	22	34	40	55	80	101
0	1	2	3	4	5	6



80

Hay que volver a  
repetir el proceso  
sobre la mitad  
derecha



# Búsqueda binaria




11	22	34	40	55	80	101
0	1	2	3	4	5	6



80

Hay que volver a  
repetir el proceso  
sobre la mitad  
derecha

# Búsqueda binaria



11	22	34	40	55	80	101
0	1	2	3	4	5	6

ítem buscado (80)


==

ítem analizado (80)



80

# Búsqueda binaria



11	22	34	40	55	80	101
0	1	2	3	4	5	6

ítem buscado (80)

==

ítem analizado (80)



80



# Búsqueda binaria



# Búsqueda binaria VS búsqueda lineal

Tendríamos 6 pasos para lograr encontrar el elemento, mientras que con solo dos pasos se resuelve con la binaria



11	22	34	40	55	80	101
0	1	2	3	4	5	6



# Búsqueda lineal VS Búsqueda binaria

$O(n)$

VS

$O(n/2)$

# Búsqueda binaria







# Ejercicios

- Escribir el algoritmo de la búsqueda lineal
- Escribir la búsqueda binaria
- Armar el algoritmo de búsqueda binaria recursiva

# Métodos de Ordenamiento

sorting

- Ordenamiento y eficiencia
- Métodos de ordenamiento sencillos
  - Orden por intercambio (método de la burbuja)
- Otros métodos de orden
  - Orden por mezcla (merge sort)
  - Ordenamiento rápido (quicksort)

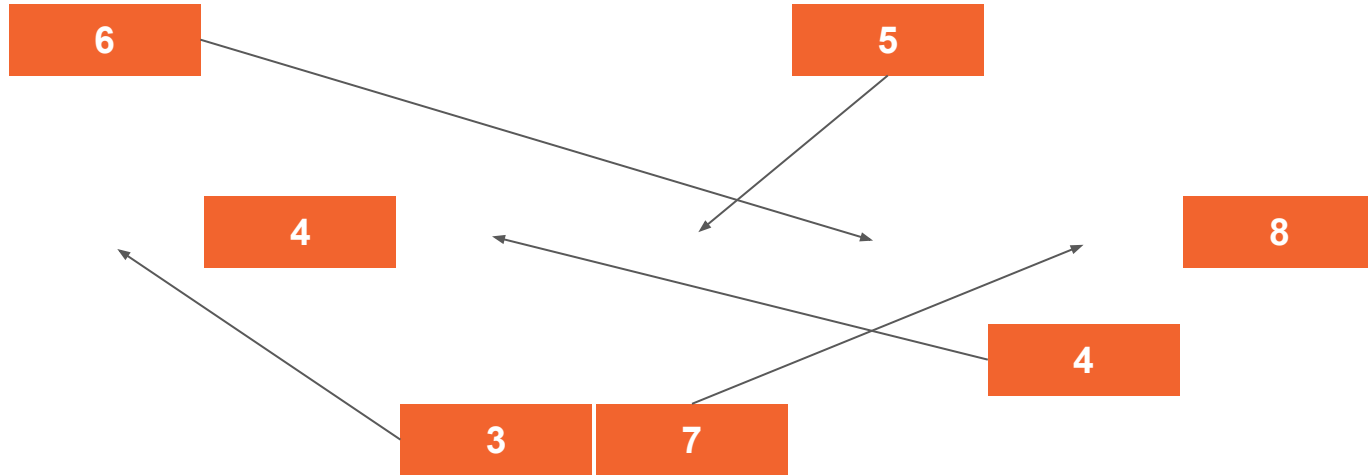
# Ordenamiento y eficiencia



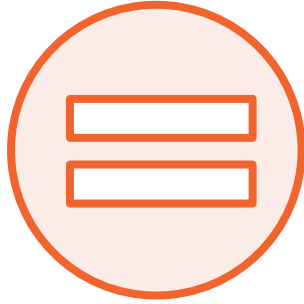
# Ordenamiento y eficiencia



# Ordenamiento y eficiencia

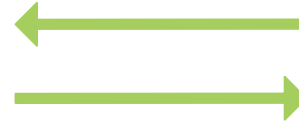


# Medir la eficiencia



## COMPARACIONES

Cuando dos elementos del array son ordenados se comparan entre sí para saber su posición relativa



## INTERCAMBIO

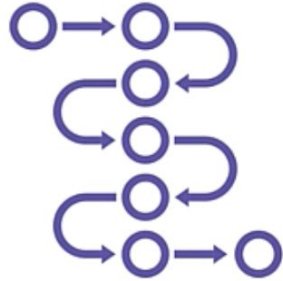
Cuando dos elementos del array son ordenados se cambian sus posiciones

# Método de la burbuja

Algoritmo por intercambio



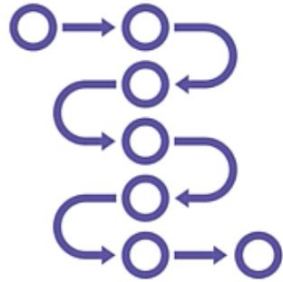
# Método de la burbuja



**Iterar**

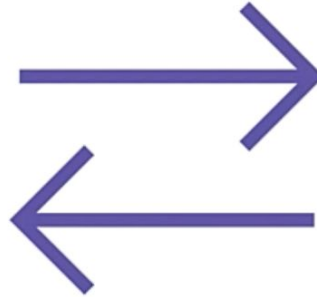
**Visitar cada ítem en el  
array desde el comienzo  
hasta el final**

# Método de la burbuja



**Iterar**

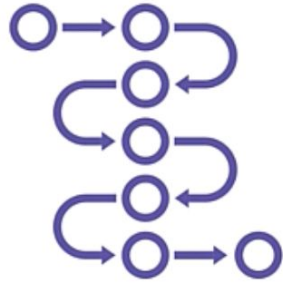
Visitar cada ítem en el  
array desde el comienzo  
hasta el final



**Intercambiar**

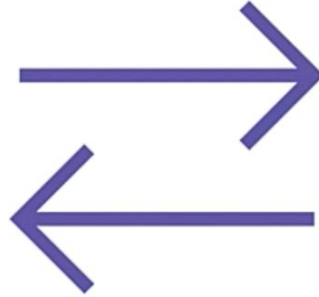
Si dos vecinos están  
fuera de orden, entonces  
se intercambian

# Método de la burbuja



**Iterar**

Visitar cada ítem en el array desde el comienzo hasta el final



**Intercambiar**

Si dos vecinos están fuera de orden, entonces se intercambian



**Repetir**

Repetir este proceso hasta que el array está ordenado

Por cada ítem

Paso 1

1	3	4	7	5	4	8
---	---	---	---	---	---	---



$$1 < 3$$

Por cada ítem

Vamos comparando

Paso 1

1	3	4	7	5	4	8
---	---	---	---	---	---	---



$$1 < 3$$

Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Paso 1

1	3	4	7	5	4	8
---	---	---	---	---	---	---



$$1 < 3$$

Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

Paso 1

1	3	4	7	5	4	8
---	---	---	---	---	---	---



$$3 < 4$$

Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

Paso 1

1	3	4	7	5	4	8
---	---	---	---	---	---	---



$$4 < 7$$



Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

Paso 1

1	3	4	7	5	4	8
---	---	---	---	---	---	---



$$7 < 5$$

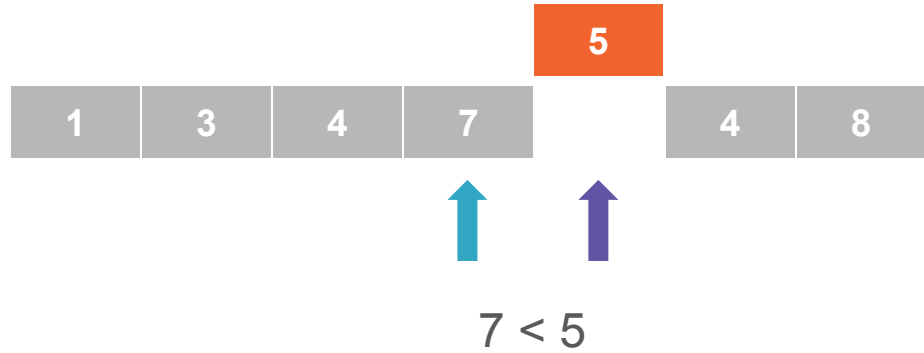
Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

Paso 1



Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

Paso 1



$$7 < 5$$

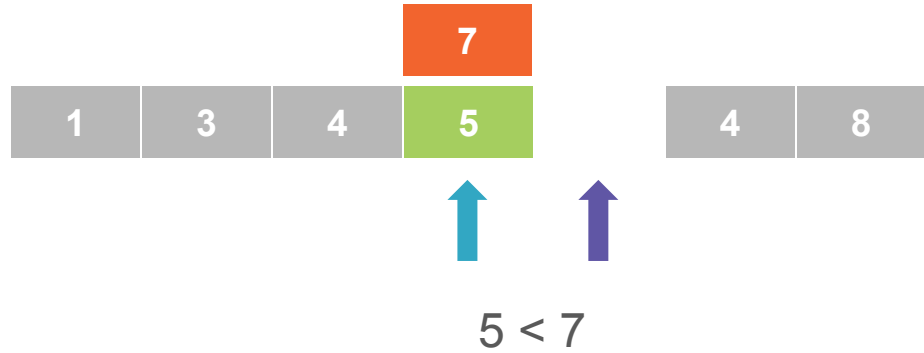
Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

Paso 1



Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

Paso 1



$$5 < 7$$

Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

Paso 1



$$7 < 4$$

Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

Paso 1



$$4 < 7$$

Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

**Repetir hasta que esté todo  
el array ordenado**

Paso 1



$7 < 8$



Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

Repetir hasta que esté todo  
el array ordenado

Paso 1

1	3	4	5	4	7	8
---	---	---	---	---	---	---

Paso 2

1	3	4	5	4	7	8
---	---	---	---	---	---	---



$$1 < 3$$

Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

**Repetir hasta que esté todo  
el array ordenado**

Paso 1

1	3	4	5	4	7	8
---	---	---	---	---	---	---

Paso 2

1	3	4	5	4	7	8
---	---	---	---	---	---	---



$$3 < 4$$

Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

Repetir hasta que esté todo  
el array ordenado

Paso 1

1	3	4	5	4	7	8
---	---	---	---	---	---	---

Paso 2

1	3	4	5	4	7	8
---	---	---	---	---	---	---



$$4 < 5$$

Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

**Repetir hasta que esté todo  
el array ordenado**

Paso 1

1	3	4	5	4	7	8
---	---	---	---	---	---	---

Paso 2

1	3	4	5	4	7	8
---	---	---	---	---	---	---



$$5 < 4$$

Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

**Repetir hasta que esté todo  
el array ordenado**

Paso 1

1	3	4	5	4	7	8
---	---	---	---	---	---	---

Paso 2

1	3	4	4	5	7	8
---	---	---	---	---	---	---



$$4 < 5$$

Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

**Repetir hasta que esté todo  
el array ordenado**

Paso 1

1	3	4	5	4	7	8
---	---	---	---	---	---	---

Paso 2

1	3	4	4	5	7	8
---	---	---	---	---	---	---



$5 < 7$

Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

**Repetir hasta que esté todo  
el array ordenado**

Paso 1

1	3	4	5	4	7	8
---	---	---	---	---	---	---

Paso 2

1	3	4	4	5	7	8
---	---	---	---	---	---	---



$7 < 8$

Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

Repetir hasta que esté todo  
el array ordenado

Paso 1

1	3	4	5	4	7	8
---	---	---	---	---	---	---

Paso 2

1	3	4	4	5	7	8
---	---	---	---	---	---	---

Paso 3

1	3	4	4	5	7	8
---	---	---	---	---	---	---





Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

Repetir hasta que esté todo  
el array ordenado

Paso 1

1	3	4	5	4	7	8
---	---	---	---	---	---	---

Paso 2

1	3	4	4	5	7	8
---	---	---	---	---	---	---

Paso 3

1	3	4	4	5	7	8
---	---	---	---	---	---	---



Por cada ítem

Vamos comparando

Intercambiamos si es mayor

Continuamos hasta el final

**Repetir hasta que esté todo  
el array ordenado**

Paso 1

1	3	4	5	4	7	8
---	---	---	---	---	---	---

Paso 2

1	3	4	4	5	7	8
---	---	---	---	---	---	---

Paso 3

1	3	4	4	5	7	8
---	---	---	---	---	---	---



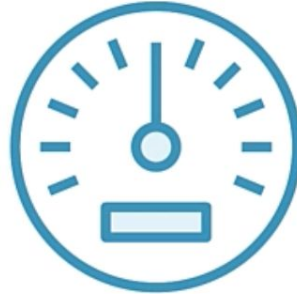
Vamos hasta el final

# Método de la burbuja y su análisis asintótico



**Mejor**

$O(n)$  comparaciones e intercambio



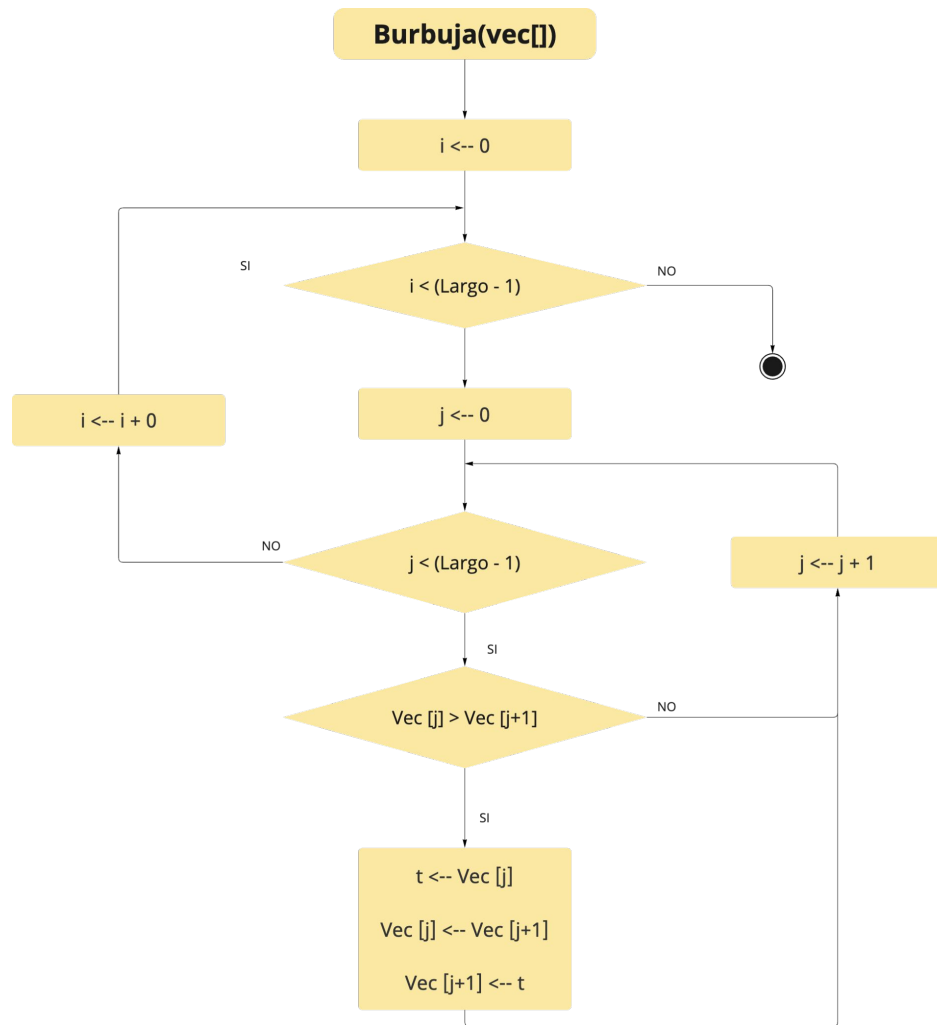
**Promedio**

$O(n^2)$  comparaciones e intercambio



**Peor**

$O(n^2)$  comparaciones e intercambio



# Ordenamiento por mezcla

## Orden por mezcla



### Cortar

Cortar el arreglo en subarreglos hasta llegar a un ítem único



### Comparar

Comparar los ítems individualmente



### Mezclar

Unir los ítems en un arreglo ordenado

**Cortar al medio cada parte**

6	4	3	7	5	4	8	2
---	---	---	---	---	---	---	---

**Cortar al medio cada parte**

6	4	3	7	5	4	8	2
---	---	---	---	---	---	---	---



**Cortar al medio cada parte**

6

4

3

7

5

4

8

2

Cortar al medio cada parte



Al llegar a este paso se supone  
que cada elemento está  
ordenado consigo mismo

**Cortar al medio cada parte**

6

4

3

7

5

4

8

2

**Cortar al medio cada parte**

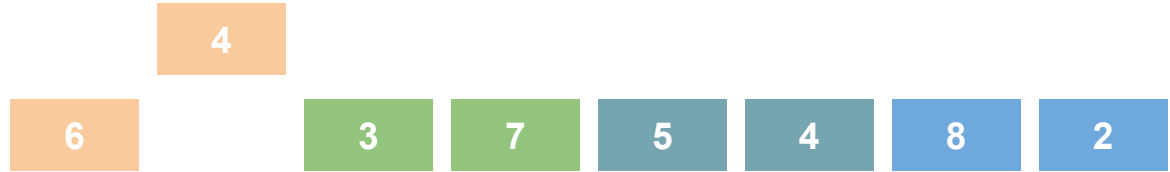
**Comparamos subarrays y  
los ordenamos en un  
nuevo array**



**Cortar al medio cada parte**

**Comparamos subarrays y  
los ordenamos en un  
nuevo array**

**Unimos (merge)**



**Cortar al medio cada parte**

**Comparamos subarrays y  
los ordenamos en un  
nuevo array**

**Unimos (merge)**



**Cortar al medio cada parte**

**Comparamos subarrays y  
los ordenamos en un  
nuevo array**

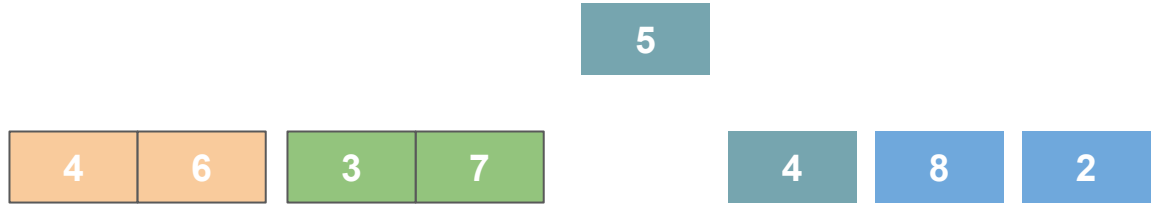
**Unimos (merge)**



Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)

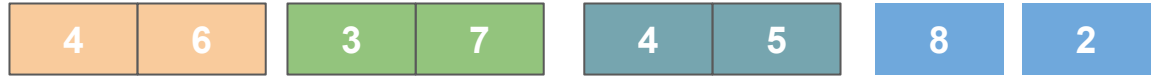




**Cortar al medio cada parte**

**Comparamos subarrays y  
los ordenamos en un  
nuevo array**

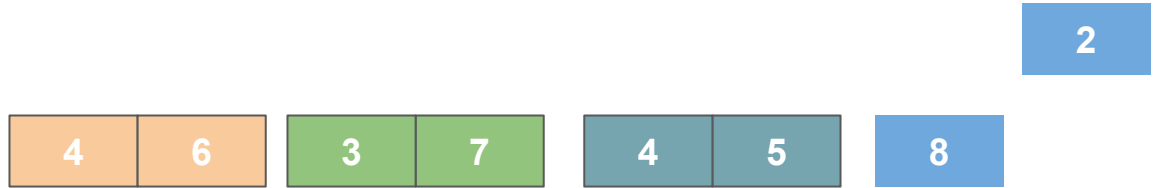
**Unimos (merge)**



**Cortar al medio cada parte**

**Comparamos subarrays y  
los ordenamos en un  
nuevo array**

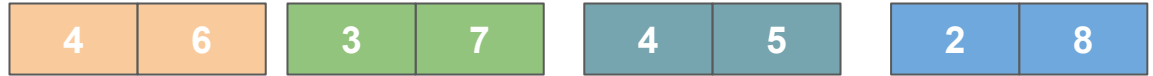
**Unimos (merge)**



**Cortar al medio cada parte**

**Comparamos subarrays y  
los ordenamos en un  
nuevo array**

**Unimos (merge)**



Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)



Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)



Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

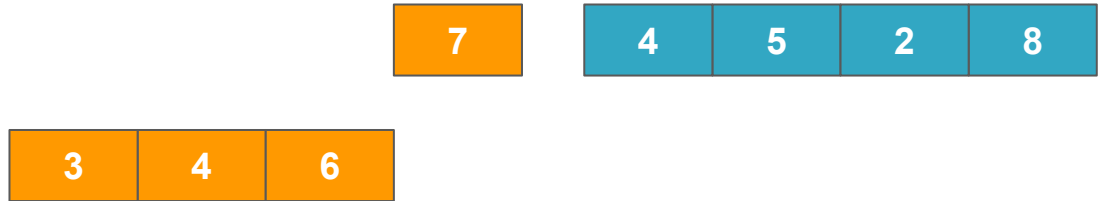
Unimos (merge)



Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)



Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)

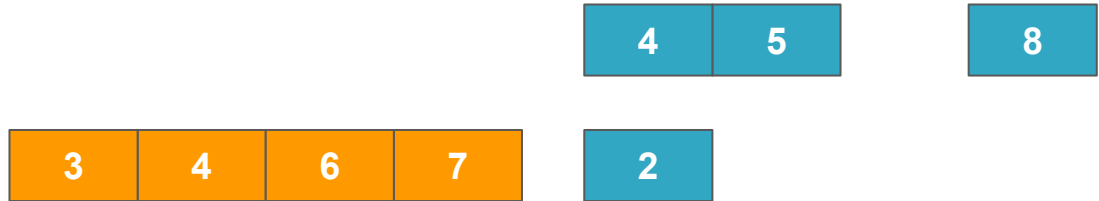




Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

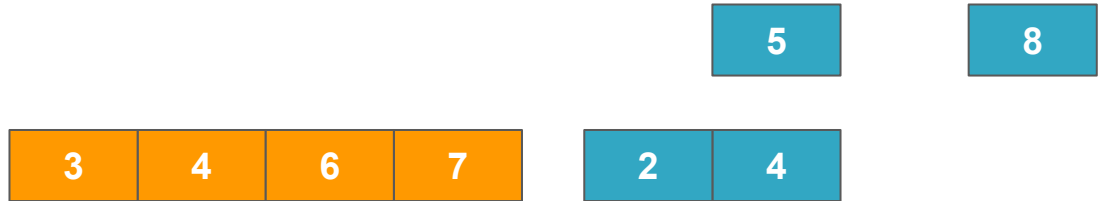
Unimos (merge)



Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)



Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)



Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)



Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)



Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)

2

3 4 6 7

4 5 8

Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)



Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)





Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)



Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)



Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)



Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

Unimos (merge)



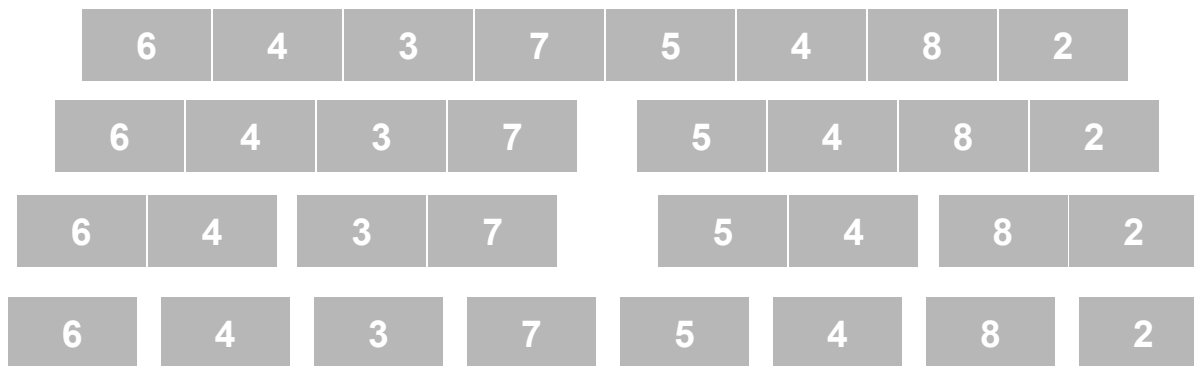
Cortar al medio cada parte

Comparamos subarrays y  
los ordenamos en un  
nuevo array

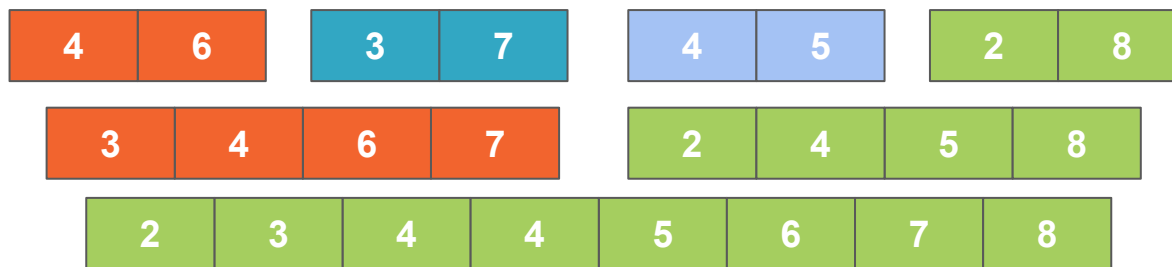
Unimos (merge)

2	3	4	4	5	6	7	8
---	---	---	---	---	---	---	---

Cortes

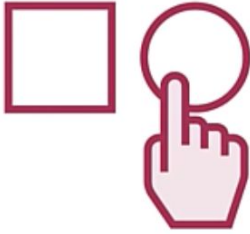


Comparar  
y Unir



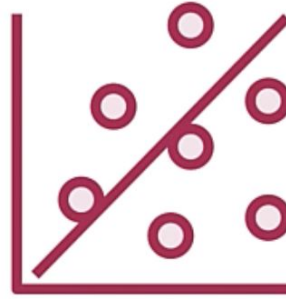
# Quicksort

# Quicksort



**Pivot**

Tomar un valor como  
pivote dentro del array



**Partition**

Reordenar los  
elementos tomando al  
pivote como referencia



**Repetir**

Repetir el método por  
cada partición del array



# ¿Qué es el pivote?

## Pivote

Es un valor de un elemento del array donde todos los valores a la izquierda son menores que él y todos los elementos de la derecha son mayores.

# Seleccionando un pivote



Seleccionar cualquier elemento dentro del array



# Seleccionando un pivote



Seleccionar cualquier elemento dentro del array



Seleccionar el primer o último elemento



# Seleccionando un pivote



Seleccionar cualquier elemento dentro del array



Seleccionar el primer o último elemento



Seleccionar el valor medio del primero, último y el medio

# Primer ítem

9	6	4	3	8	5	4	1	2	6
---	---	---	---	---	---	---	---	---	---

# Primer ítem

9	6	4	3	8	5	4	1	2	6
---	---	---	---	---	---	---	---	---	---

# Primer último

9	6	4	3	8	5	4	1	2	6
---	---	---	---	---	---	---	---	---	---

# Problemas al seleccionar primero o último



Requiere  $O(n^2)$  operaciones en un caso donde esté pre ordenado



# Valor Random

9	6	4	3	8	5	4	1	2	6
---	---	---	---	---	---	---	---	---	---

# El valor del medio de los tres



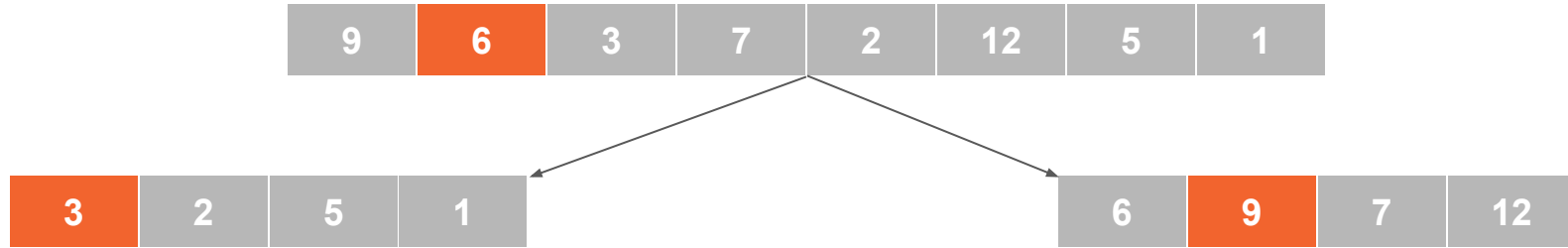
# El valor del medio de los tres

9	6	4	3	8	5	4	1	2	6
---	---	---	---	---	---	---	---	---	---

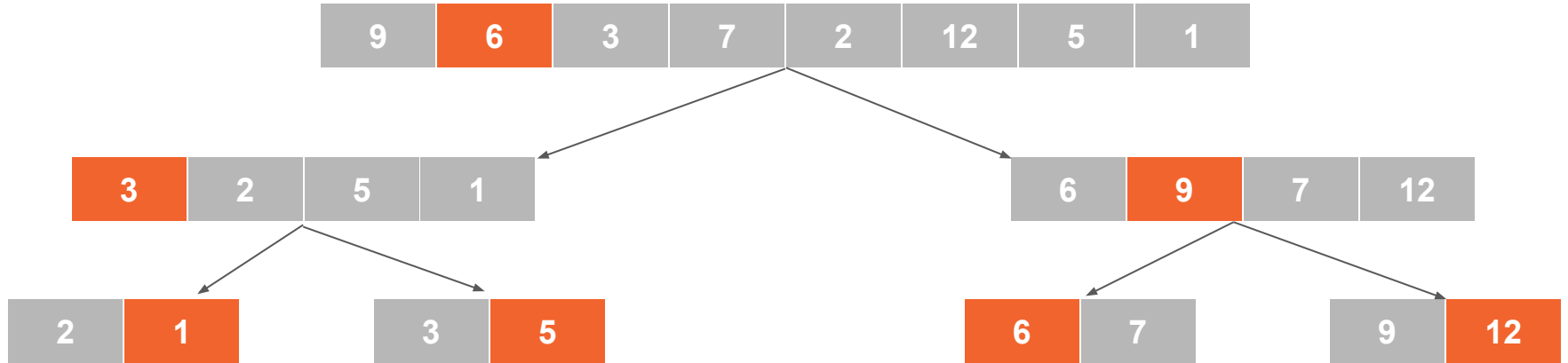
# Un valor random

9	6	3	7	2	12	5	1
---	---	---	---	---	----	---	---

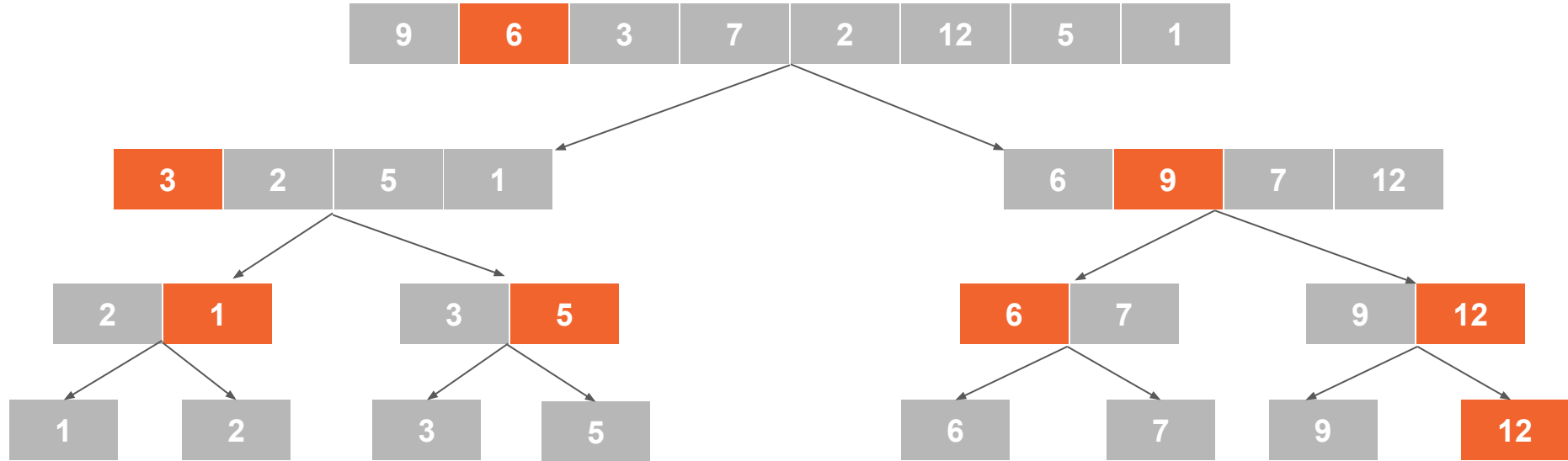
# Un valor random



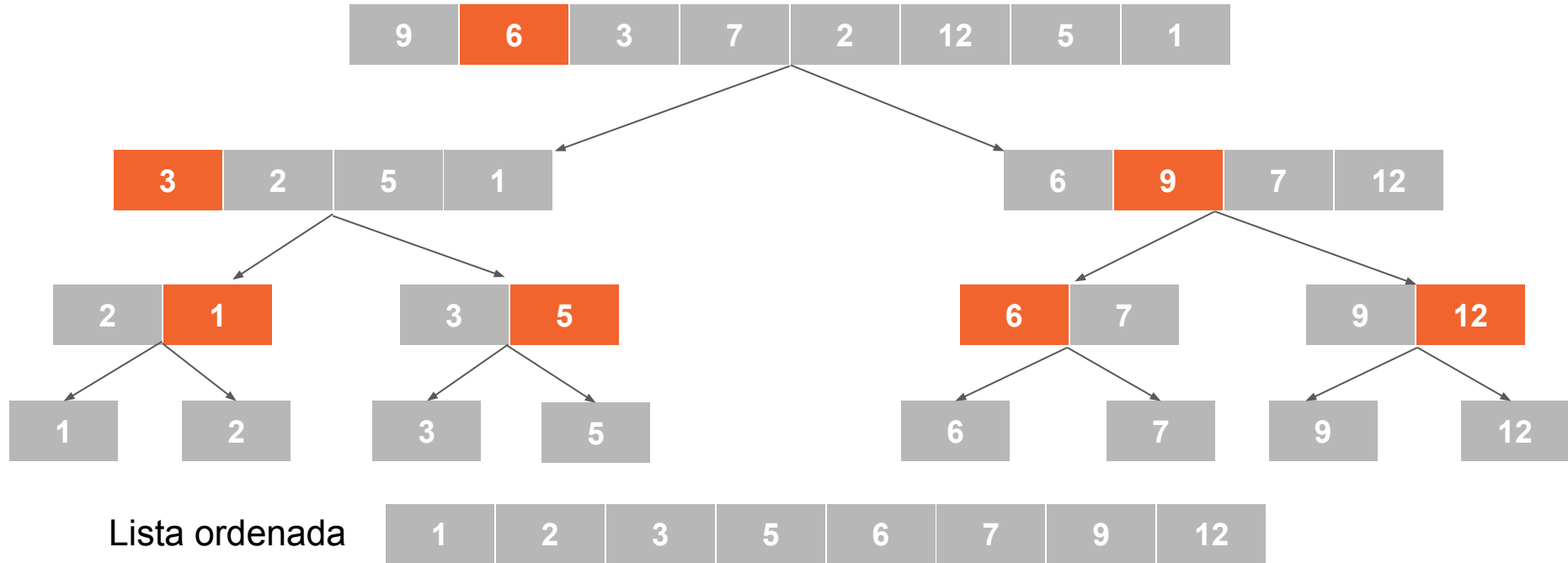
# Un valor random



# Un valor random



# Un valor random





# Propiedades del algoritmo Quicksort



## Divide y Conquista

Reduce  
el problema  
al caso  
más particular



## Ajustado

Solo requiere  
de  $O(\log n)$  de  
utilización  
de la memoria



## Optimizable

Existen muchas  
optimizaciones  
para mejorar  
la performance

