

Algorítmica y Programación I

ARREGLOS -
ORDENAMIENTO



Algoritmo de Ordenamiento

- › Proceso por el cual un grupo de elementos se puede ordenar de acuerdo a cierto criterio.
- › EJ: criterio : ordenados de menor a mayor

Arreglo desordenado – no cumple el criterio establecido

10	5	-3	6	0
----	---	----	---	---

Arreglo ordenado – cumple el criterio establecido

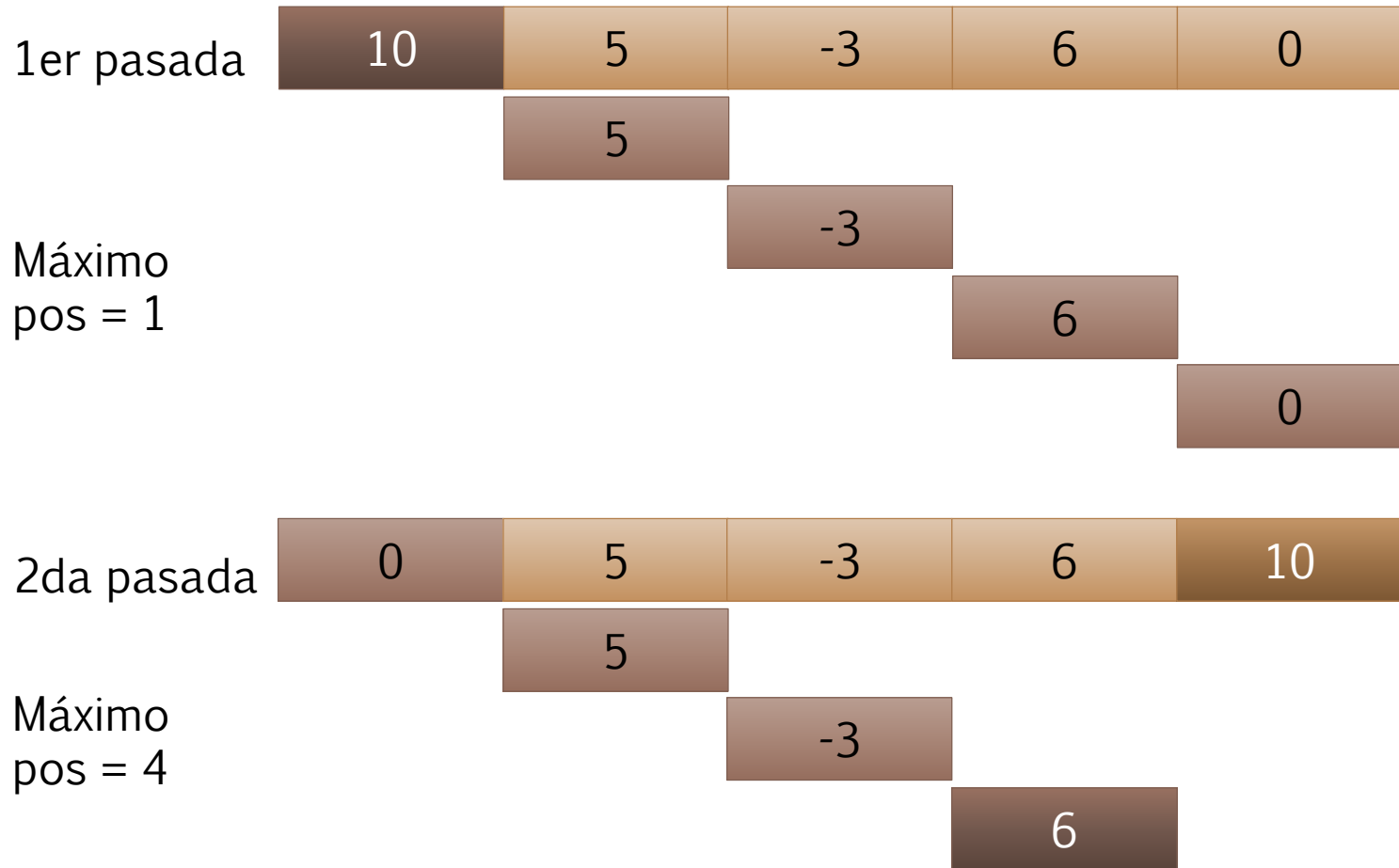
-3	0	5	6	10
----	---	---	---	----

Método ordenamiento - Selección

- › Consiste en **buscar el elemento mayor** del arreglo y **posicionarlo al final**. Luego se procede a realizar la misma operación pero teniendo en cuenta que se recorre desde el principio hasta el anteúltimo ya que el último ha quedado ordenado en la pasada anterior. Se continúan las recorridas hasta posicionar todos los elementos.
- › Dependiendo del criterio con el que se quiera ordenar se buscará el mayor o el menor.
- › Se puede realizar el algoritmo intercambiando al principio en vez del final.

Método ordenamiento – Selección

criterio: de menor a mayor

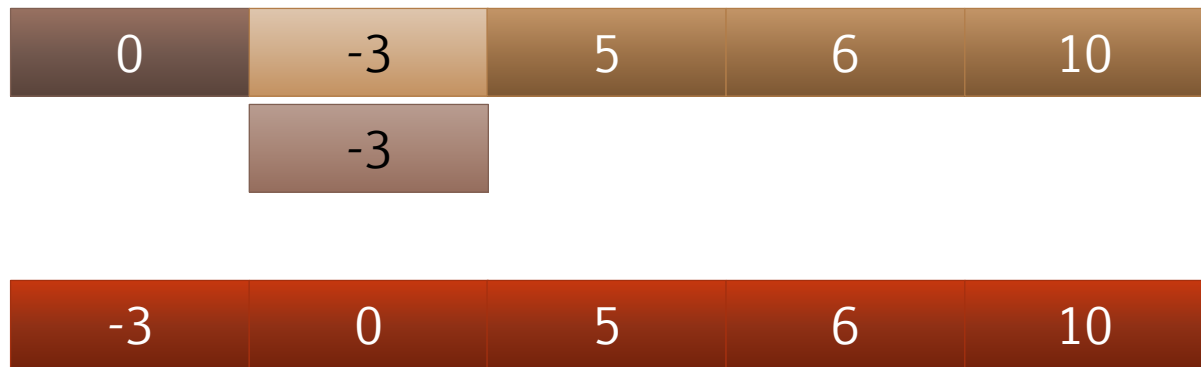


Método ordenamiento - Selección

3er pasada



4ta pasada



Método ordenamiento – Intercambio o Burbuja

- › El método realiza $n-1$ pasadas (al igual que el método de selección). En cada pasada se comparan los elementos adyacentes y se intercambian si están desordenados.
- › Se lo conoce como método de burbuja porque se dice que los elementos pequeños o de menor peso burbujan hacia arriba (inicio).
- › Una mejora del método consiste en verificar si hubo intercambio al finalizar una pasada, en caso de no producirse el intercambio se termina ya que puede garantizarse que el arreglo se encuentra ordenado.

Método ordenamiento - Burbuja

1er pasada

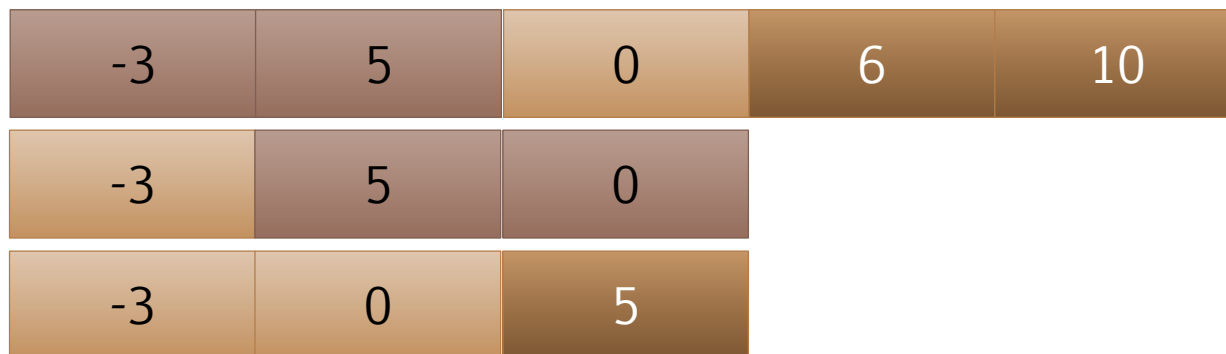
10	5	-3	6	0
5	10	-3		
5	-3	10	6	
5	-3	6	10	0

2da pasada

5	-3	6	0	10
-3	5	6		
-3	5	6	0	
-3	5	0	6	

Método ordenamiento - Burbuja

3ra pasada



4ta pasada

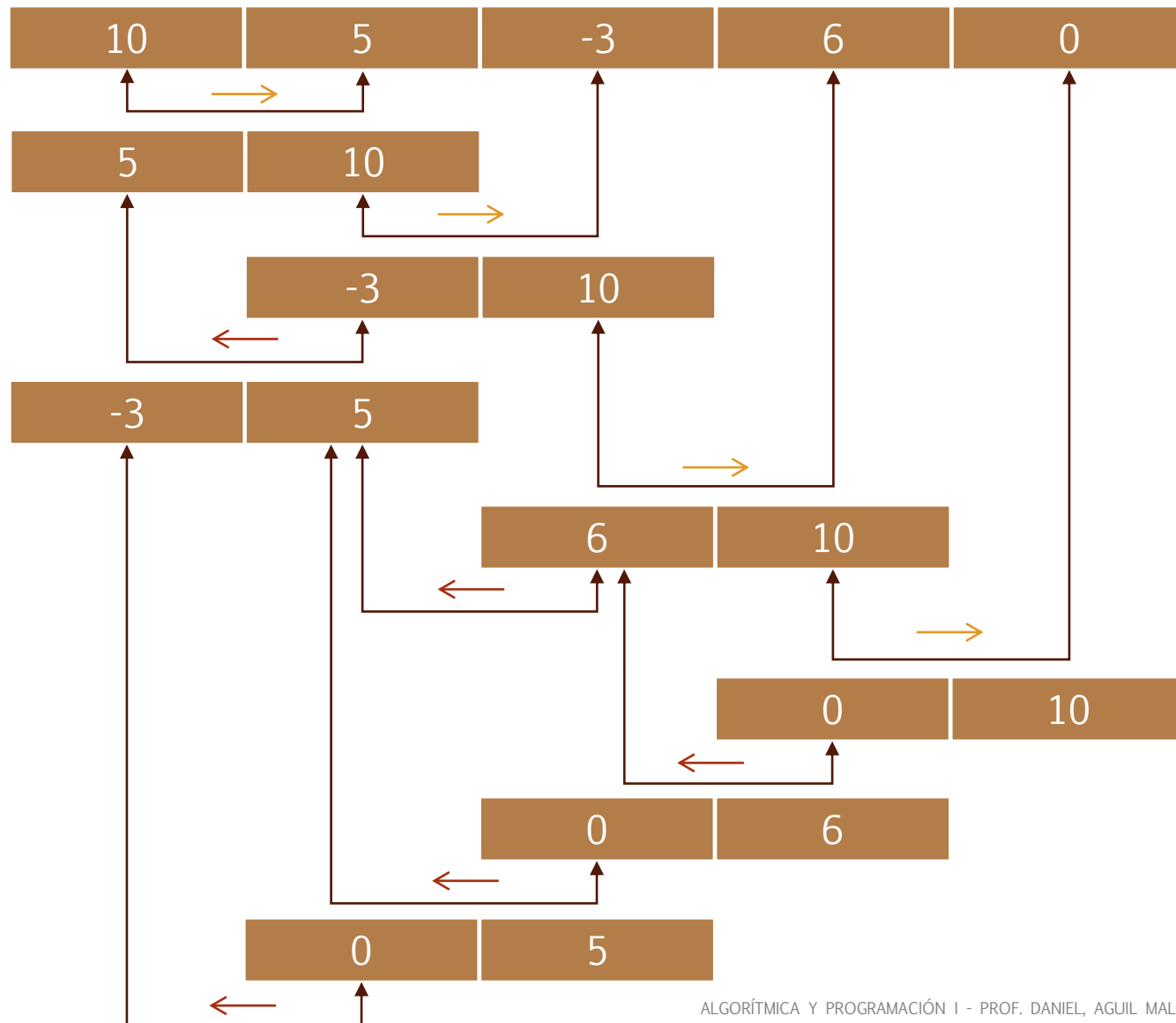


Método ordenamiento - Inserción

- › El método consiste en recorrer los elementos del arreglo desde la posición 2 hasta el final. Se verifica que cada elemento se encuentre ordenado con respecto al elemento anterior y si esto no sucede se lleva a su posición correcta intercambiando los elementos en su recorrido (hacia atrás).
- › Otra forma de describir el método: recorrer todo el arreglo A comenzando desde la posición $pos=2$ y terminando en $pos=N$. Para cada pos , se trata de ubicar en el lugar correcto el elemento $A[pos]$ entre los elementos anteriores: $A[pos-1]$, $A[pos-2]$, ..., $A[1]$.

Método ordenamiento - inserción

π



Estrategia ordenamiento indexado

- › Los métodos visto anteriormente son ineficientes con respecto de otros métodos pero sirven como punto inicial para su estudio posterior.
- › El orden de estos métodos es de N^2 salvo en situaciones especiales.
- › Con un arreglo semi-ordenado se puede concluir que el método de la burbuja (mejorada) o el método por inserción funcionan con un orden igual a N .
- › Una estrategia que suele utilizarse cuando se ordenan arreglos con tipos de datos complejos, ya que se degrada el tiempo de respuesta del método al realizarse los intercambios, es utilizar un arreglo índice.

Estrategia ordenamiento indexado

› Ej ordenamiento por legajo

V		Arreglo			
Pos		Legajo	Nombre	Apellido	Etc....
1	→	1010	Jose	Garcia	...
2	→	2401	Ariel	Perez	...
...	
N	→	980	Tomás	Torres	...

```
If( Arreglo[ V[i] ].legajo > Arreglo[ V[i+1] ].legajo ) then  
    intercambio(V[i], V[i+1])
```

Pos		Legajo	Nombre	Apellido	Etc....
2	↗	1010	Jose	Garcia	...
N	↘	2401	Ariel	Perez	...
...	
1	↖	980	Tomás	Torres	...

Mezcla

- › Supongamos que contamos con los nombres de los alumnos de dos cursos diferentes (A y B), ¿Cómo podríamos hacer para imprimir un listado de ambos cursos ordenados por apellido (C) ?

A=

perez	garcia	gomez	aguiar	tapia
-------	--------	-------	--------	-------

 N = 5

B=

mendez	gutierrez	aguiar
--------	-----------	--------

 M = 3

$$N + M = 8$$

C=

aguiar	aguiar	garcia	gomez	gutierrez	mendez	perez	tapia
--------	--------	--------	-------	-----------	--------	-------	-------

- ```

if (A[i] <= B[j]) then
 C[k] := A[i]
else
 C[k] := B[j]
k := k + 1

```