

Algoritmos II

Fundamentos de Computación

Diego Caro

José Fernandez

Fernanda Kri

2020-2

Resultados de aprendizaje

- Describir el funcionamiento de algoritmos recursivos
- Describir mecanismo funcionamiento algoritmos de ordenación
- Comprender conceptos de tiempo de ejecución, peor caso y mejor caso

Recordatorio: funciones

- Secuencia de instrucciones diseñadas para resolver una **tarea** específica.
- El objetivo es reutilizarla en programas (o pseudocódigo) cada vez que se desea resolver esa **tarea**.
- La función está definida por un nombre, una entrada, una salida, y la secuencia de instrucciones.
 - La entrada es lo que deseamos procesar (pueden ser números, texto, lista de elementos, etc...)
 - La salida es el resultado de computar las instrucciones con la entrada.
 - La salida se especifica con la palabra **retornar**.
- Es análogo a funciones matemáticas!

Función en matemáticas

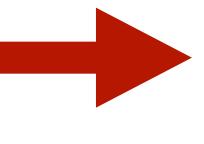
$$f(x) = x^2$$

Función en pseudocódigo

$f(x)$:
1. **retornar** $x*x$

**¿Como calcular el número
máximo para N números?**

1. Sea X una lista $X_1, X_2, \dots, X_N; X_i \in \mathbb{R}$
2. Leer N números y dejarlos en la lista X
3. $\text{max} = \text{Maximo}(X)$
4. Imprimir “El máximo numero en X es”, max

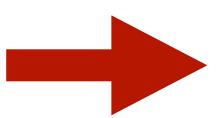


X	max
5,9,7	-

$\text{Maximo}(Z)$:

1. Sea $m = -\infty$ un número real negativo
2. **Para $i = 1$ hasta N :**
3. Si $Z_i > m$:
4. $m = Z_i$
5. **retornar m**

1. Sea X una lista $X_1, X_2, \dots, X_N; X_i \in \mathbb{R}$
2. Leer N números y dejarlos en la lista X
- 3. $\max = \text{Maximo}(X)$**
4. Imprimir “El máximo numero en X es”, \max



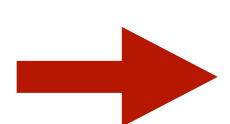
X	\max
5,9,7	-
5,9,7	?????

$\text{Maximo}(Z)$:

1. Sea $m = -\infty$ un número real negativo
2. **Para $i = 1$ hasta N :**
3. Si $Z_i > m$:
4. $m = Z_i$
5. **retornar** m

1. Sea X una lista $X_1, X_2, \dots, X_N; X_i \in \mathbb{R}$
2. Leer N números y dejarlos en la lista X
- 3. $\max = \text{Maximo}(X)$**
4. Imprimir “El máximo numero en X es”, \max

X	max
5,9,7	-

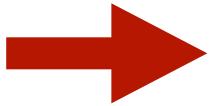


Maximo(Z):

1. Sea $m = -\infty$ un número real negativo
2. **Para $i = 1$ hasta N :**
3. Si $Z_i > m$:
4. $m = Z_i$
5. **retornar m**

Z	m	i	Z _i	Z _i > m
5,9,7	-	-	-	-

1. Sea X una lista $X_1, X_2, \dots, X_N; X_i \in \mathbb{R}$
2. Leer N números y dejarlos en la lista X
3. $\max = \text{Maximo}(X)$
4. Imprimir “El máximo numero en X es”, \max



X	\max
5,9,7	-
5,9,7	?????

- Maximo(Z):** 
1. Sea $m = -\infty$ un número real negativo
 2. **Para** $i = 1$ **hasta** N :
 3. Si $Z_i > m$:
 4. $m = Z_i$
 5. **retornar** m



Z	m	i	Z_i	$Z_i > m$
5,9,7	-	-	-	-
5,9,7	$-\infty$	-	-	-

1. Sea X una lista $X_1, X_2, \dots, X_N; X_i \in \mathbb{R}$
2. Leer N números y dejarlos en la lista X
3. $\max = \text{Maximo}(X)$
4. Imprimir “El máximo numero en X es”, \max

X	\max
5,9,7	-
5,9,7	?????

- Maximo(Z):
1. Sea $m = -\infty$ un número real negativo
 2. **Para $i = 1$ hasta N :**
 3. Si $Z_i > m$:
 4. $m = Z_i$
 5. **retornar m**

Z	m	i	Z_i	$Z_i > m$
5,9,7	-	-	-	-
5,9,7	$-\infty$	-	-	-
5,9,7	$-\infty$	1	5	

1. Sea X una lista $X_1, X_2, \dots, X_N; X_i \in \mathbb{R}$
2. Leer N números y dejarlos en la lista X
3. $\max = \text{Maximo}(X)$
4. Imprimir “El máximo numero en X es”, \max

X	max
5,9,7	-
5,9,7	?????

- Maximo(Z):
1. Sea $m = -\infty$ un número real negativo
 2. **Para** $i = 1$ **hasta** N :
 3. Si $Z_i > m$:
 4. $m = Z_i$
 5. **retornar** m

Z	m	i	Z _i	Z _i > m
5,9,7	-	-	-	-
5,9,7	-∞	-	-	-
5,9,7	-∞	1	5	V

1. Sea X una lista $X_1, X_2, \dots, X_N; X_i \in \mathbb{R}$
2. Leer N números y dejarlos en la lista X
3. $\max = \text{Maximo}(X)$
4. Imprimir “El máximo numero en X es”, \max

X	max
5,9,7	-
5,9,7	?????

- Maximo(Z):
1. Sea $m = -\infty$ un número real negativo
 2. **Para** $i = 1$ **hasta** N :
 3. Si $Z_i > m$:
 4. $m = Z_i$
 5. **retornar** m

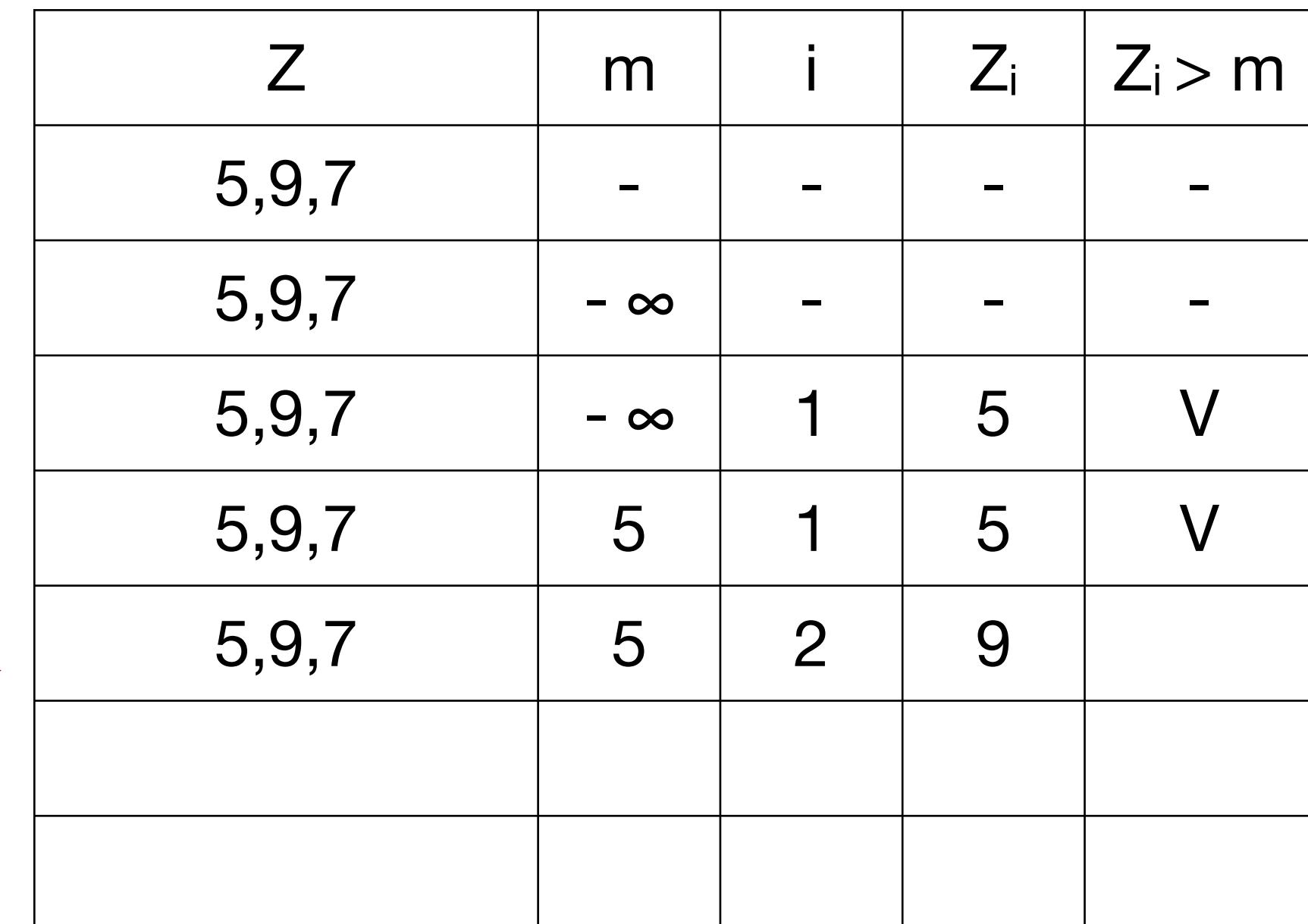
Z	m	i	Z_i	$Z_i > m$
5,9,7	-	-	-	-
5,9,7	$-\infty$	-	-	-
5,9,7	$-\infty$	1	5	v
5,9,7	5	1	5	v

1. Sea X una lista $X_1, X_2, \dots, X_N; X_i \in \mathbb{R}$
2. Leer N números y dejarlos en la lista X
3. $\max = \text{Maximo}(X)$
4. Imprimir “El máximo numero en X es”, \max



X	\max
5,9,7	-
5,9,7	?????

- Maximo(Z):
1. Sea $m = -\infty$ un número real negativo
 2. **Para $i = 1$ hasta N :**
 3. Si $Z_i > m$:
 4. $m = Z_i$
 5. **retornar m**



Z	m	i	Z_i	$Z_i > m$
5,9,7	-	-	-	-
5,9,7	$-\infty$	-	-	-
5,9,7	$-\infty$	1	5	V
5,9,7	5	1	5	V
5,9,7	5	2	9	

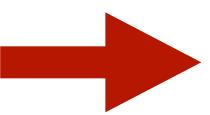
1. Sea X una lista $X_1, X_2, \dots, X_N; X_i \in \mathbb{R}$
2. Leer N números y dejarlos en la lista X
3. $\max = \text{Maximo}(X)$
4. Imprimir “El máximo numero en X es”, \max

X	max
5,9,7	-
5,9,7	?????

- Maximo(Z):
1. Sea $m = -\infty$ un número real negativo
 2. **Para $i = 1$ hasta N :**
 3. Si $Z_i > m$:
 4. $m = Z_i$
 5. **retornar m**

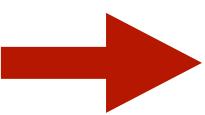
Z	m	i	Z _i	Z _i > m
5,9,7	-	-	-	-
5,9,7	- ∞	-	-	-
5,9,7	- ∞	1	5	V
5,9,7	5	1	5	V
5,9,7	5	2	9	V

1. Sea X una lista $X_1, X_2, \dots, X_N; X_i \in \mathbb{R}$
2. Leer N números y dejarlos en la lista X
3. $\max = \text{Maximo}(X)$
4. Imprimir “El máximo numero en X es”, \max



X	\max
5,9,7	-
5,9,7	?????

- Maximo(Z):
1. Sea $m = -\infty$ un número real negativo
 2. **Para** $i = 1$ **hasta** N :
 3. Si $Z_i > m$:
 4. $m = Z_i$
 5. **retornar** m



Z	m	i	Z_i	$Z_i > m$
5,9,7	-	-	-	-
5,9,7	$-\infty$	-	-	-
5,9,7	$-\infty$	1	5	V
5,9,7	5	1	5	V
5,9,7	5	2	9	V
5,9,7	9	2	9	V

1. Sea X una lista $X_1, X_2, \dots, X_N; X_i \in \mathbb{R}$
2. Leer N números y dejarlos en la lista X
3. $\max = \text{Maximo}(X)$
4. Imprimir “El máximo numero en X es”, \max

X	max
5,9,7	-
5,9,7	?????

- Maximo(Z):
1. Sea $m = -\infty$ un número real negativo
 2. **Para** $i = 1$ **hasta** N :
 3. Si $Z_i > m$:
 4. $m = Z_i$
 5. **retornar** m

Z	m	i	Z _i	Z _i > m
5,9,7	-	-	-	-
5,9,7	-∞	-	-	-
5,9,7	-∞	1	5	V
5,9,7	5	1	5	V
5,9,7	5	2	9	V
5,9,7	9	2	9	V
5,9,7	9	3	7	F

1. Sea X una lista $X_1, X_2, \dots, X_N; X_i \in \mathbb{R}$
2. Leer N números y dejarlos en la lista X
3. $\max = \text{Maximo}(X)$
4. Imprimir “El máximo numero en X es”, \max

X	max
5,9,7	-
5,9,7	?????

- Maximo(Z):
1. Sea $m = -\infty$ un número real negativo
 2. **Para** $i = 1$ **hasta** N :
 3. Si $Z_i > m$:
 4. $m = Z_i$
 5. **retornar** m

Z	m	i	Z _i	Z _i > m
5,9,7	-	-	-	-
5,9,7	-∞	-	-	-
5,9,7	-∞	1	5	V
5,9,7	5	1	5	V
5,9,7	5	2	9	V
5,9,7	9	2	9	V
5,9,7	9	3	7	F

1. Sea X una lista $X_1, X_2, \dots, X_N; X_i \in \mathbb{R}$
2. Leer N números y dejarlos en la lista X
3. $\text{max} = \text{Maximo}(X)$
- 4. Imprimir “El máximo numero en X es”, max**

→

X	max
5,9,7	-
5,9,7	9

→

$\text{Maximo}(Z)$:

1. Sea $m = -\infty$ un número real negativo
2. **Para $i = 1$ hasta N :**
3. Si $Z_i > m$:
4. $m = Z_i$
- 5. retornar m**



1. Sea X una lista $X_1, X_2, \dots, X_N; X_i \in \mathbb{R}$
2. Leer N números y dejarlos en la lista X
3. $\text{max} = \text{Maximo}(X)$
- 4. Imprimir “El máximo numero en X es”, max**

→

X	max
5,9,7	-
5,9,7	9

→

$\text{Maximo}(Z)$:

1. Sea $m = -\infty$ un número real negativo
2. **Para $i = 1$ hasta N :**
3. Si $Z_i > m$:
4. $m = Z_i$
- 5. retornar m**



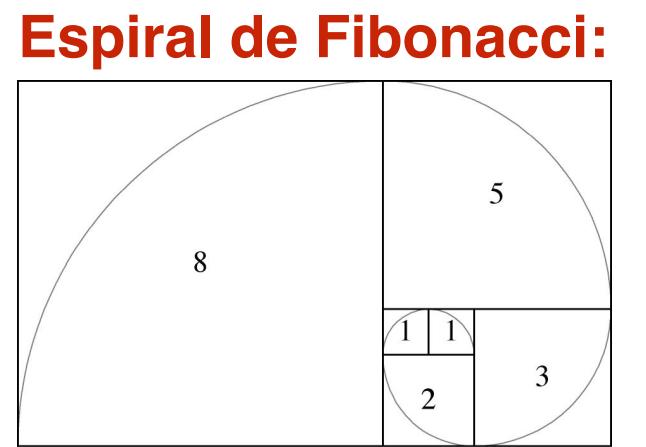
Funciones recursivas

- Algoritmo que calcula una salida de forma recursiva, es decir, ejecutándose a si mismo.
- Leonardo Fibonacci, matemático italiano conocido por su famosa secuencia:
 - 0, 1, 1, 2, 3, 5, 8, 13, 21, 34...
- Definición recursiva:

¿Cuánto es F(3)?

$$F_n = \begin{cases} F_{n-1} + F_{n-2} & \text{if } n > 1 \\ 1 & \text{if } n = 1 \\ 0 & \text{if } n = 0. \end{cases}$$

$$F(3) = F(2) + F(1)$$



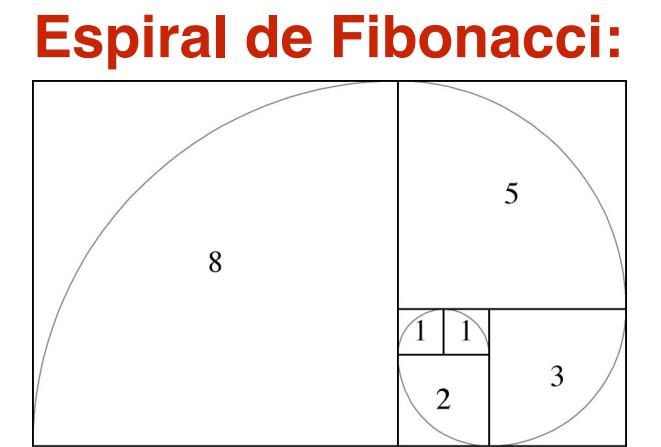
Funciones recursivas

- Algoritmo que calcula una salida de forma recursiva, es decir, ejecutándose a si mismo.
- Leonardo Fibonacci, matemático italiano conocido por su famosa secuencia:
 - 0, 1, 1, 2, 3, 5, 8, 13, 21, 34...
- Definición recursiva:

¿Cuánto es F(3)?

$$F_n = \begin{cases} F_{n-1} + F_{n-2} & \text{if } n > 1 \\ 1 & \text{if } n = 1 \\ 0 & \text{if } n = 0. \end{cases}$$

$$\begin{aligned} F(3) &= F(2) + F(1) \\ F(2) &= F(1) + F(0) \end{aligned}$$



Funciones recursivas

- Algoritmo que calcula una salida de forma recursiva, es decir, ejecutándose a si mismo.
- Leonardo Fibonacci, matemático italiano conocido por su famosa secuencia:
 - 0, 1, 1, 2, 3, 5, 8, 13, 21, 34...
- Definición recursiva:

¿Cuánto es F(3)?

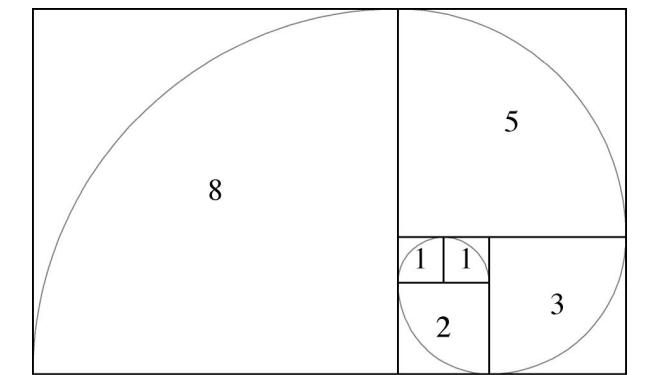
$$F_n = \begin{cases} F_{n-1} + F_{n-2} & \text{if } n > 1 \\ 1 & \text{if } n = 1 \\ 0 & \text{if } n = 0. \end{cases}$$

$$F(3) = F(2) + F(1)$$

$$F(2) = F(1) + F(0)$$

$$F(1) = 1$$

Espiral de Fibonacci:



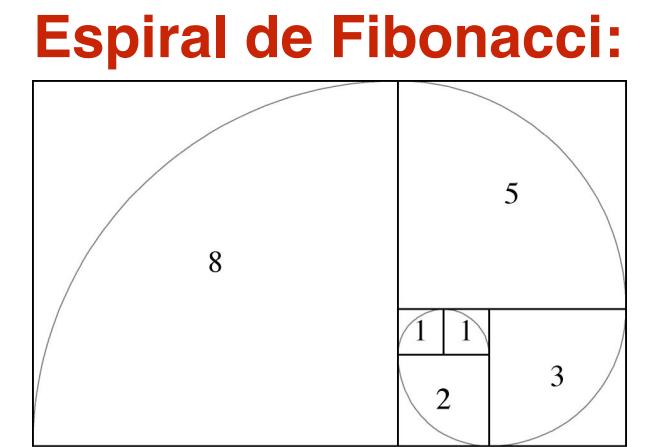
Funciones recursivas

- Algoritmo que calcula una salida de forma recursiva, es decir, ejecutándose a si mismo.
- Leonardo Fibonacci, matemático italiano conocido por su famosa secuencia:
 - 0, 1, 1, 2, 3, 5, 8, 13, 21, 34...
- Definición recursiva:

¿Cuánto es F(3)?

$$F_n = \begin{cases} F_{n-1} + F_{n-2} & \text{if } n > 1 \\ 1 & \text{if } n = 1 \\ 0 & \text{if } n = 0. \end{cases}$$

$$\begin{aligned} F(3) &= F(2) + F(1) \\ &\quad + F(0) \\ F(2) &= 1 \\ F(1) &= 1 \end{aligned}$$



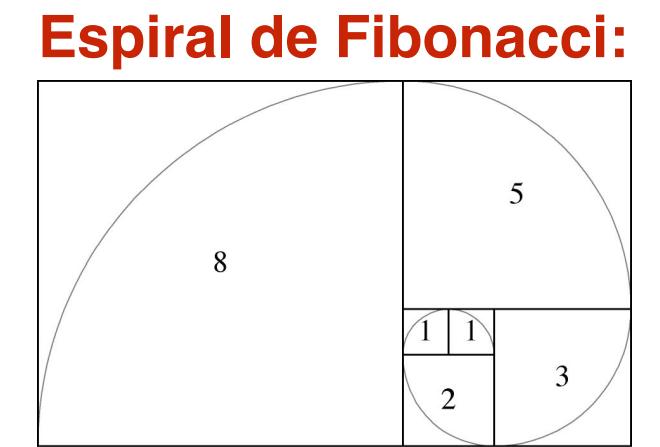
Funciones recursivas

- Algoritmo que calcula una salida de forma recursiva, es decir, ejecutándose a si mismo.
- Leonardo Fibonacci, matemático italiano conocido por su famosa secuencia:
 - 0, 1, 1, 2, 3, 5, 8, 13, 21, 34...
- Definición recursiva:

¿Cuánto es F(3)?

$$F_n = \begin{cases} F_{n-1} + F_{n-2} & \text{if } n > 1 \\ 1 & \text{if } n = 1 \\ 0 & \text{if } n = 0. \end{cases}$$

$$\begin{aligned} F(3) &= F(2) + F(1) \\ &+ F(0) \\ F(2) &= 1 \\ F(1) &= 1 \\ F(0) &= 0 \end{aligned}$$



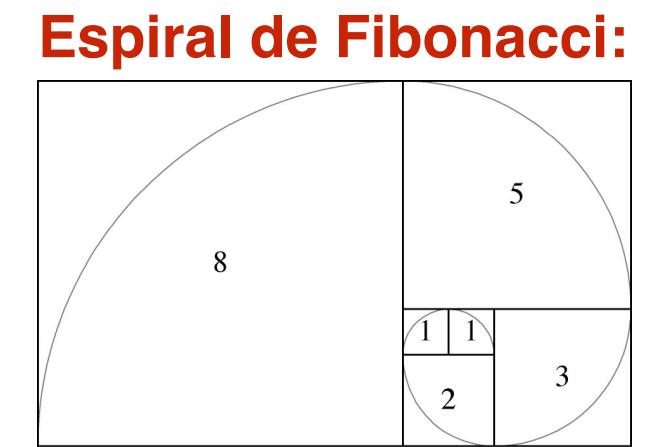
Funciones recursivas

- Algoritmo que calcula una salida de forma recursiva, es decir, ejecutándose a si mismo.
- Leonardo Fibonacci, matemático italiano conocido por su famosa secuencia:
 - 0, 1, 1, 2, 3, 5, 8, 13, 21, 34...
- Definición recursiva:

¿Cuánto es F(3)?

$$F_n = \begin{cases} F_{n-1} + F_{n-2} & \text{if } n > 1 \\ 1 & \text{if } n = 1 \\ 0 & \text{if } n = 0. \end{cases}$$

$$\begin{aligned} F(3) &= F(2) + F(1) \\ F(2) &= 1 \\ F(1) &= 1 \\ F(0) &= 0 \end{aligned}$$



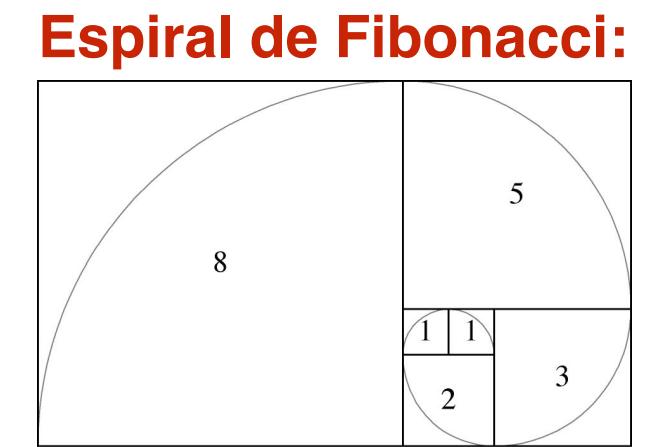
Funciones recursivas

- Algoritmo que calcula una salida de forma recursiva, es decir, ejecutándose a si mismo.
- Leonardo Fibonacci, matemático italiano conocido por su famosa secuencia:
 - 0, 1, 1, 2, 3, 5, 8, 13, 21, 34...
- Definición recursiva:

¿Cuánto es F(3)?

$$F_n = \begin{cases} F_{n-1} + F_{n-2} & \text{if } n > 1 \\ 1 & \text{if } n = 1 \\ 0 & \text{if } n = 0. \end{cases}$$

$$\begin{aligned} F(3) &= 1 & + F(1) \\ F(2) &= 1 & + 0 = 1 \\ F(1) &= 1 \\ F(0) &= 0 \end{aligned}$$

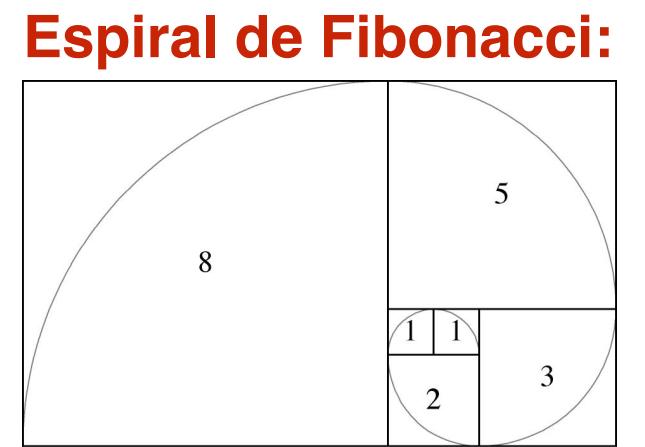
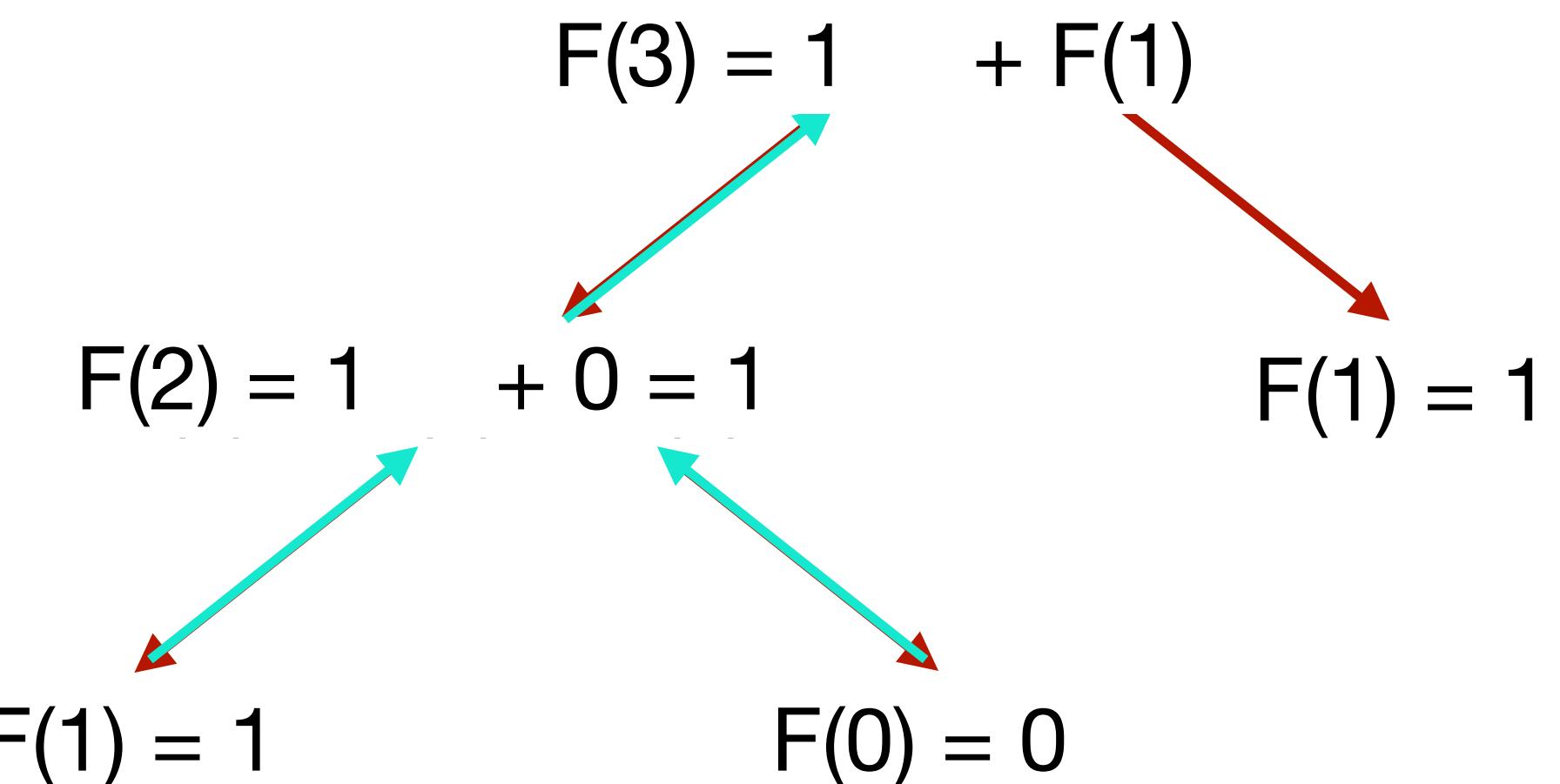


Funciones recursivas

- Algoritmo que calcula una salida de forma recursiva, es decir, ejecutándose a si mismo.
- Leonardo Fibonacci, matemático italiano conocido por su famosa secuencia:
 - 0, 1, 1, 2, 3, 5, 8, 13, 21, 34...
- Definición recursiva:

¿Cuánto es F(3)?

$$F_n = \begin{cases} F_{n-1} + F_{n-2} & \text{if } n > 1 \\ 1 & \text{if } n = 1 \\ 0 & \text{if } n = 0. \end{cases}$$



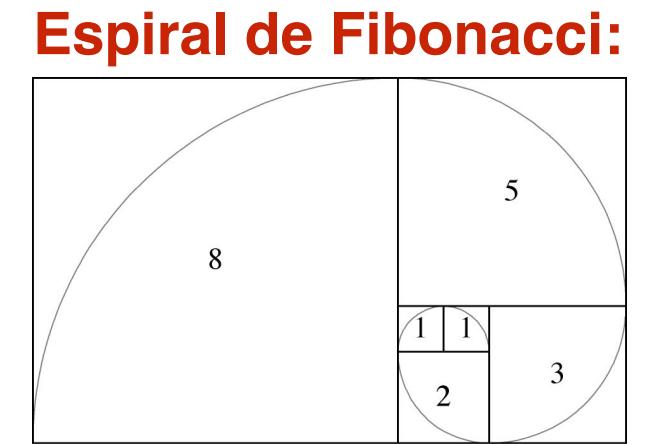
Funciones recursivas

- Algoritmo que calcula una salida de forma recursiva, es decir, ejecutándose a si mismo.
- Leonardo Fibonacci, matemático italiano conocido por su famosa secuencia:
 - 0, 1, 1, 2, 3, 5, 8, 13, 21, 34...
- Definición recursiva:

¿Cuánto es $F(3)$?

$$F_n = \begin{cases} F_{n-1} + F_{n-2} & \text{if } n > 1 \\ 1 & \text{if } n = 1 \\ 0 & \text{if } n = 0. \end{cases}$$

$$\begin{array}{ccccccc} F(3) = 1 & + 1 & = 2 \\ \downarrow & & \downarrow \\ F(2) = 1 & + 0 = 1 & & & & & \\ \downarrow & & \downarrow & & & & \\ F(1) = 1 & & & & & & \\ \downarrow & & & & & & \\ F(0) = 0 & & & & & & \end{array}$$



Funciones recursivas

- Cómo construir una función recursiva: definir casos
 - **Caso Recursivo:** ¿con qué valores la función se llama a si misma?
 - **Caso Base:** ¿con qué valores se detiene la función?
- Puede haber más de un caso base, y más de un caso recursivo.
- Debes asegurarte que la función termine con al menos un caso base.

$$F_n = \begin{cases} F_{n-1} + F_{n-2} & \text{if } n > 1 \\ 1 & \text{if } n = 1 \\ 0 & \text{if } n = 0. \end{cases}$$

Caso recursivo

Caso base

Caso base

$F(n)$:

1. **Si** $n > 1$:
2. **retornar** $F(n-1) + F(n-2)$
3. **Si** $n == 1$:
4. **retornar** 1
5. **Si** $n == 0$:
6. **retornar** 0

$$\text{Factorial}(n) = \begin{cases} n \times \text{Factorial}(n - 1) & \text{si } n > 1 \\ 1 & \text{si } n == 1 \end{cases}$$

Caso recursivo
Caso base

$$\text{Factorial}(3) = 3 * \text{Factorial}(2)$$

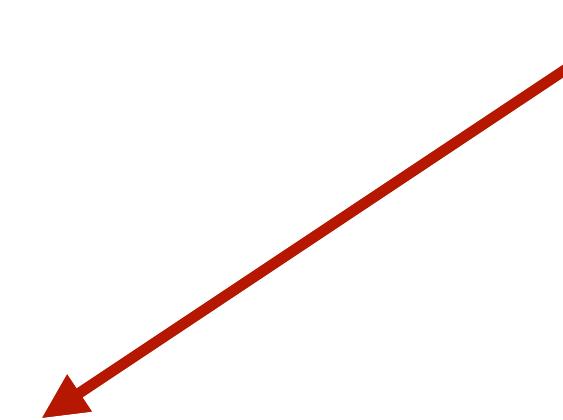
Factorial(n):

1. **Si** $n > 1$: **Caso recursivo**
2. **retornar** $n * \text{Factorial}(n-1)$
3. **Si** $n == 1$: **Caso base**
4. **retornar** 1

$$\text{Factorial}(n) = \begin{cases} n \times \text{Factorial}(n - 1) & \text{si } n > 1 \\ 1 & \text{si } n == 1 \end{cases}$$

Caso recursivo
Caso base

$$\text{Factorial}(3) = 3 * \text{Factorial}(2)$$



$$\text{Factorial}(2) = 2 * \text{Factorial}(1)$$

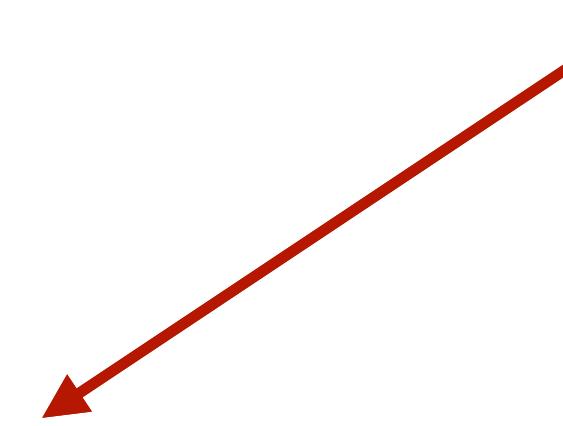
Factorial(n):

1. **Si** $n > 1$: **Caso recursivo**
2. **retornar** $n * \text{Factorial}(n-1)$
3. **Si** $n == 1$: **Caso base**
4. **retornar** 1

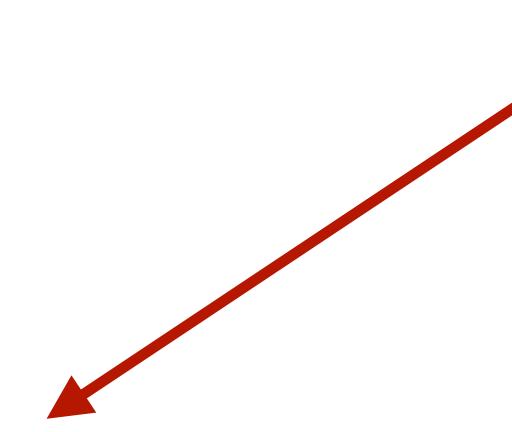
$$\text{Factorial}(n) = \begin{cases} n \times \text{Factorial}(n - 1) & \text{si } n > 1 \\ 1 & \text{si } n == 1 \end{cases}$$

Caso recursivo
Caso base

$$\text{Factorial}(3) = 3 * \text{Factorial}(2)$$



$$\text{Factorial}(2) = 2 * \text{Factorial}(1)$$



$$\text{Factorial}(1) = 1$$

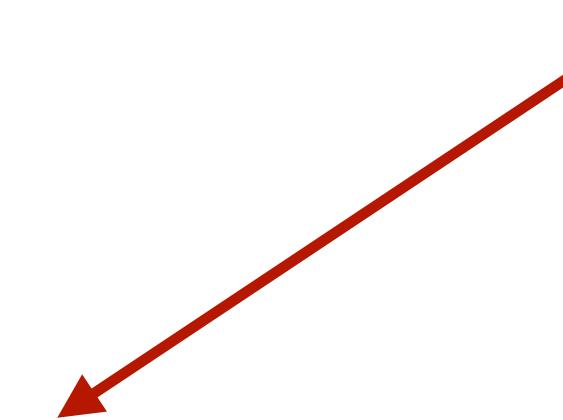
Factorial(n):

1. **Si** $n > 1$: **Caso recursivo**
2. **retornar** $n * \text{Factorial}(n-1)$
3. **Si** $n == 1$: **Caso base**
4. **retornar** 1

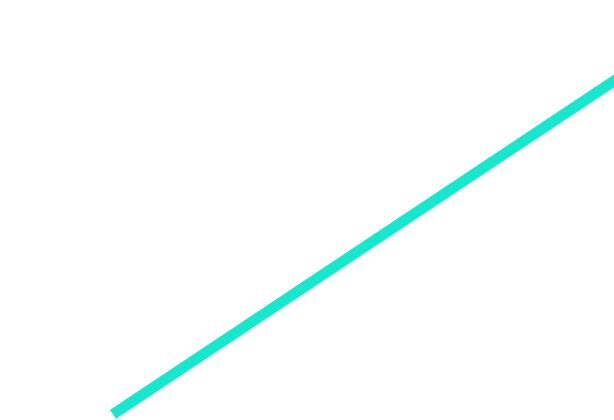
$$\text{Factorial}(n) = \begin{cases} n \times \text{Factorial}(n - 1) & \text{si } n > 1 \\ 1 & \text{si } n == 1 \end{cases}$$

Caso recursivo
Caso base

$$\text{Factorial}(3) = 3 * \text{Factorial}(2)$$



$$\text{Factorial}(2) = 2 * 1 = 2$$



$$\text{Factorial}(1) = 1$$

Factorial(n):

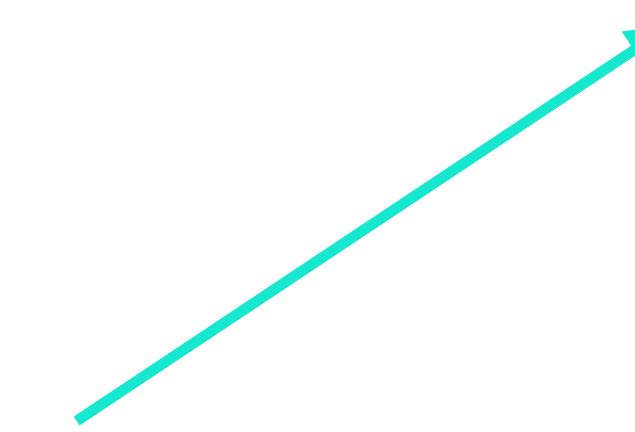
1. **Si** $n > 1$: **Caso recursivo**
2. **retornar** $n * \text{Factorial}(n-1)$
3. **Si** $n == 1$: **Caso base**
4. **retornar** 1

$$\text{Factorial}(n) = \begin{cases} n \times \text{Factorial}(n - 1) & \text{si } n > 1 \\ 1 & \text{si } n == 1 \end{cases}$$

Caso recursivo

Caso base

$$\text{Factorial}(3) = 3 * 2 = 6$$



$$\text{Factorial}(2) = 2 * 1 = 2$$

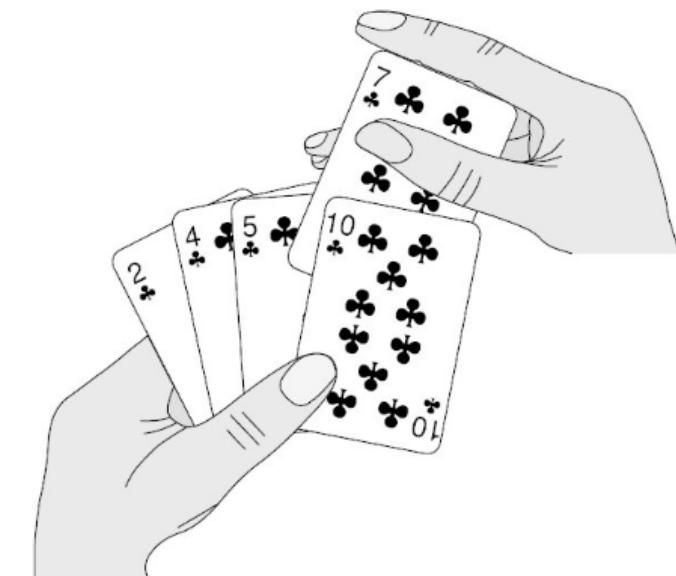
Factorial(n):

1. **Si** $n > 1$: Caso recursivo
2. **retornar** $n * \text{Factorial}(n-1)$
3. **Si** $n == 1$: Caso base
4. **retornar** 1

Ordenamiento por Inserción

- Idea: estrategia similar al ordenar cartas para formar una escala en el juego de naipes **carioca**.
- Es un algoritmo iterativo,
 - Mantiene una sublista de elementos ordenados en cada iteración.
 - Inserta un nuevo elemento, y lo mueve hasta hacer que la sublista se mantenga ordenada.
 - Y así sucesivamente hasta que se ha insertado el último elemento de la lista.

a: 6 5 3 1 8 7 2 4



Ordenamiento por Inserción

- Idea: estrategia similar al ordenar cartas para formar una escala en el juego de naipes **carioca**.
- Es un algoritmo iterativo,
 - Mantiene una sublista de elementos ordenados en cada iteración.
 - Inserta un nuevo elemento, y lo mueve hasta hacer que la sublista se mantenga ordenada.
 - Y así sucesivamente hasta que se ha insertado el último elemento de la lista.

a: 6 5 3 1 8 7 2 4

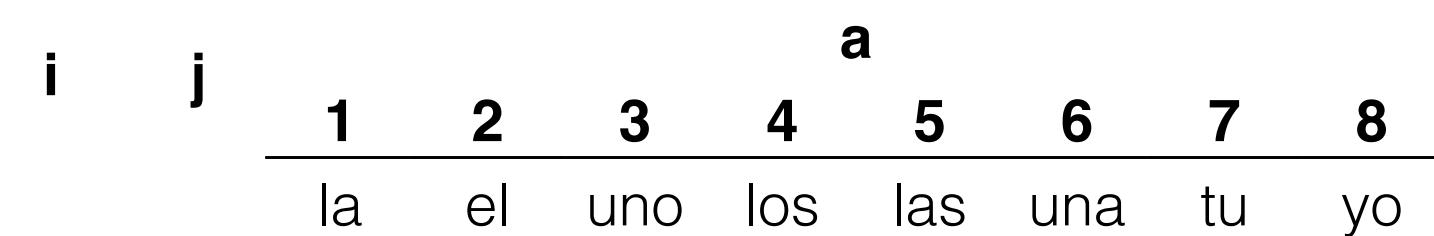


Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

Atención: asuma que el texto se compara siguiendo orden lexicográfico (tal como está en un diccionario).

la	el	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

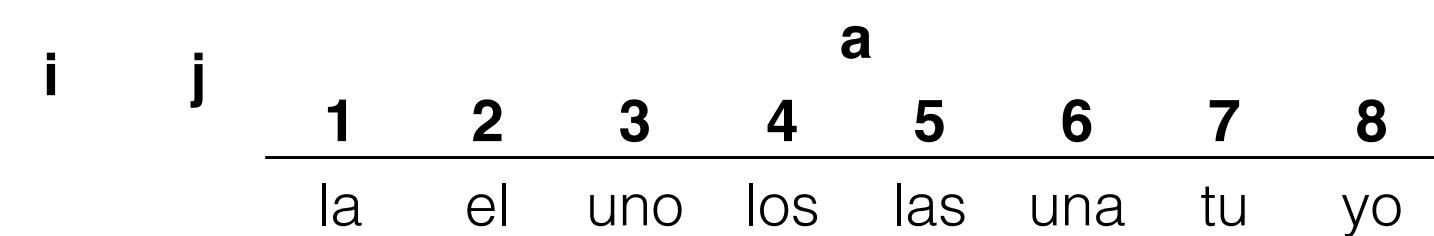
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

Atención: asuma que el texto se compara siguiendo orden lexicográfico (tal como está en un diccionario).

la	el	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

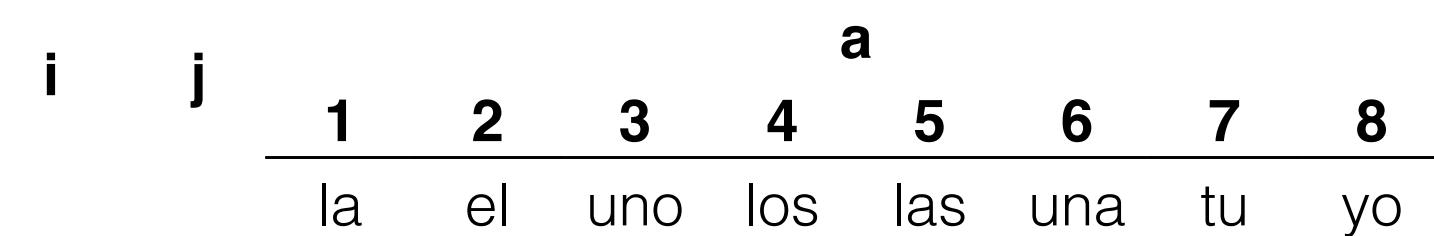
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

Atención: asuma que el texto se compara siguiendo orden lexicográfico (tal como está en un diccionario).

la	el	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

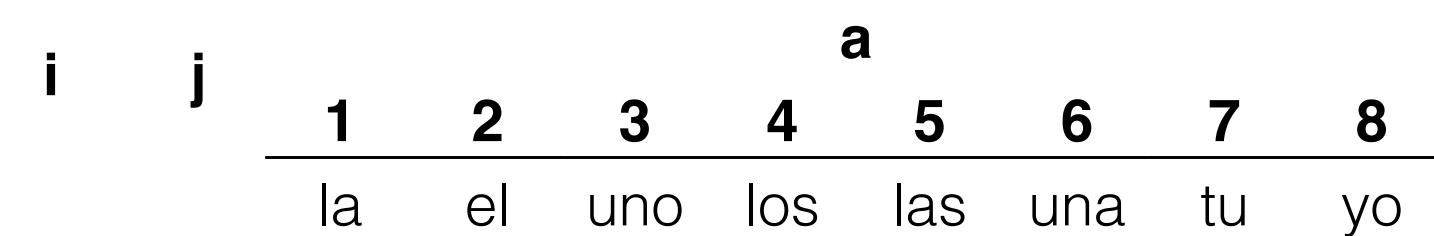
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

Atención: asuma que el texto se compara siguiendo orden lexicográfico (tal como está en un diccionario).

la	el	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

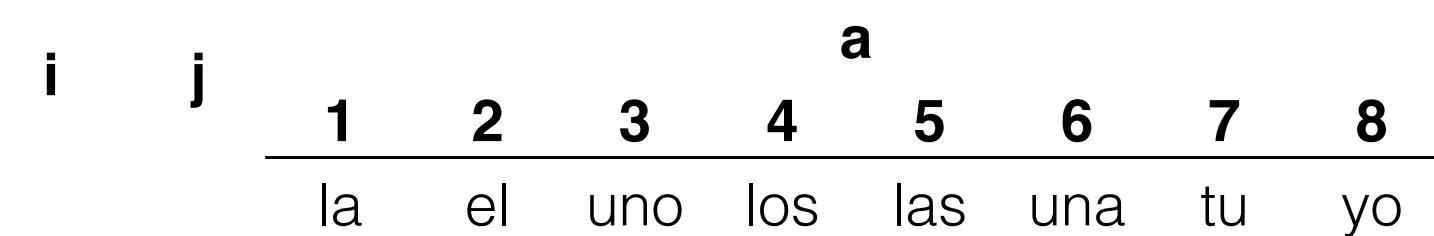
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

Atención: asuma que el texto se compara siguiendo orden lexicográfico (tal como está en un diccionario).

la	el	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

Atención: asuma que el texto se compara siguiendo orden lexicográfico (tal como está en un diccionario).

la	el	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8

i	j	a	1	2	3	4	5	6	7	8
			la	el	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

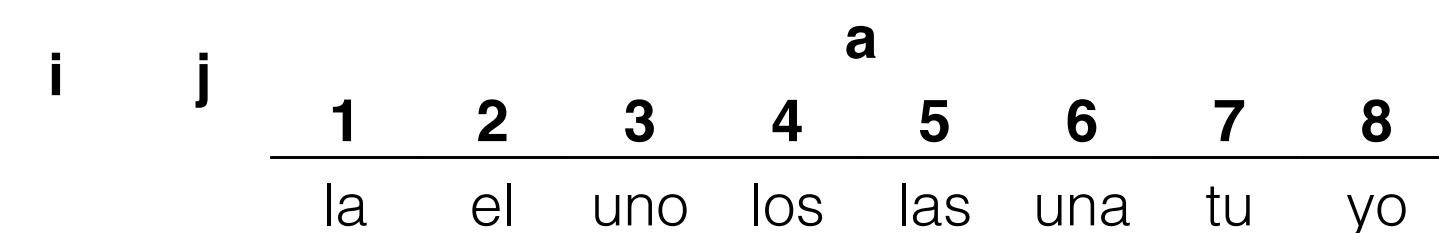
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

Atención: asuma que el texto se compara siguiendo orden lexicográfico (tal como está en un diccionario).

la	el	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

Atención: asuma que el texto se compara siguiendo orden lexicográfico (tal como está en un diccionario).

la	el	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8

i	j	a	1	2	3	4	5	6	7	8
			la	el	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

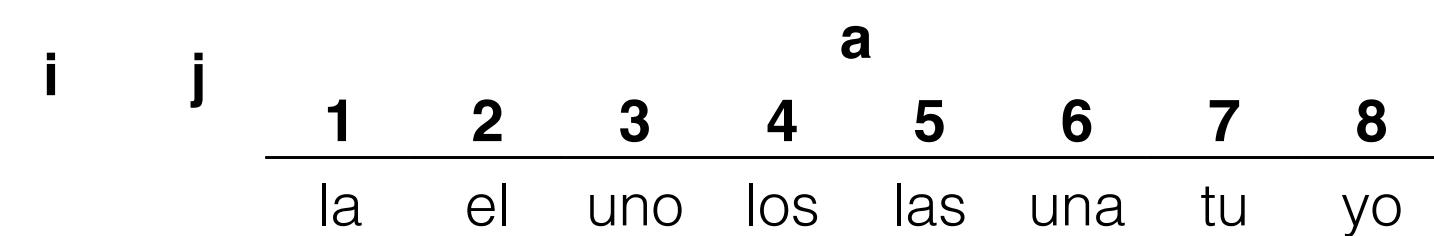
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

Atención: asuma que el texto se compara siguiendo orden lexicográfico (tal como está en un diccionario).

la	el	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

Atención: asuma que el texto se compara siguiendo orden lexicográfico (tal como está en un diccionario).

la	el	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8

i	j	a	1	2	3	4	5	6	7	8
			la	el	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

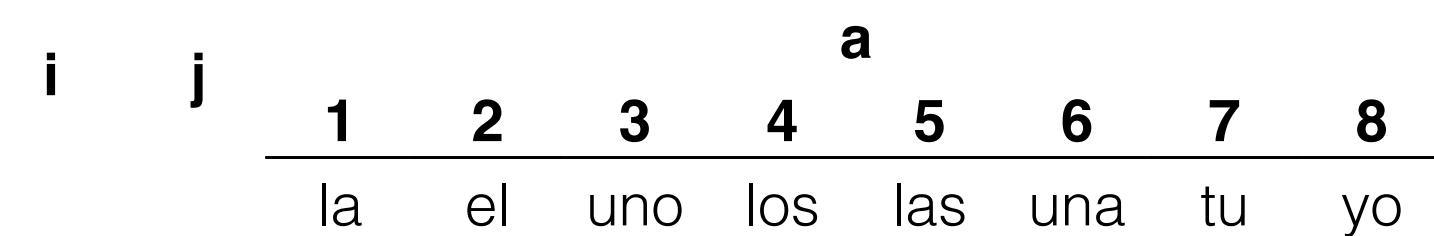
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

Atención: asuma que el texto se compara siguiendo orden lexicográfico (tal como está en un diccionario).

la	el	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

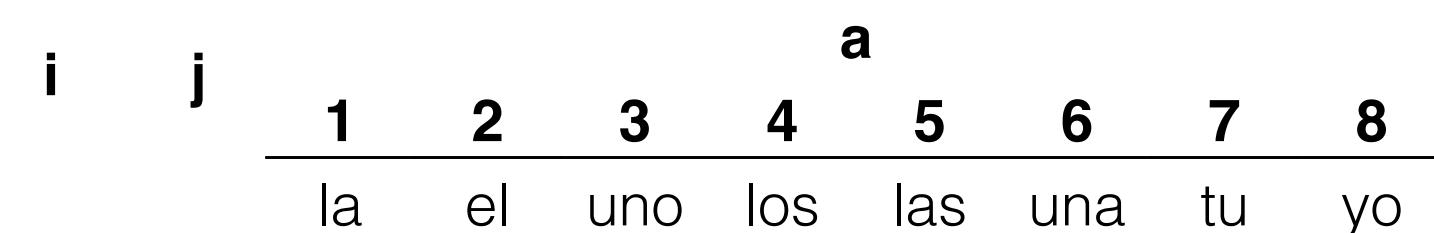
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

Atención: asuma que el texto se compara siguiendo orden lexicográfico (tal como está en un diccionario).

la	el	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

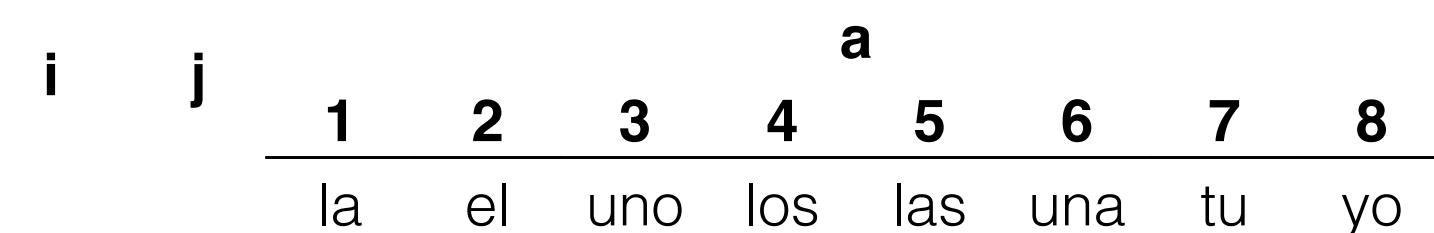
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

Atención: asuma que el texto se compara siguiendo orden lexicográfico (tal como está en un diccionario).

la	el	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

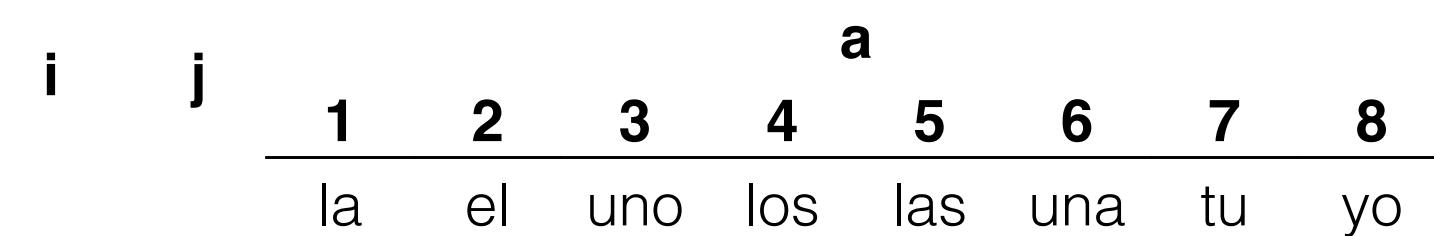
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

Atención: asuma que el texto se compara siguiendo orden lexicográfico (tal como está en un diccionario).

la	el	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

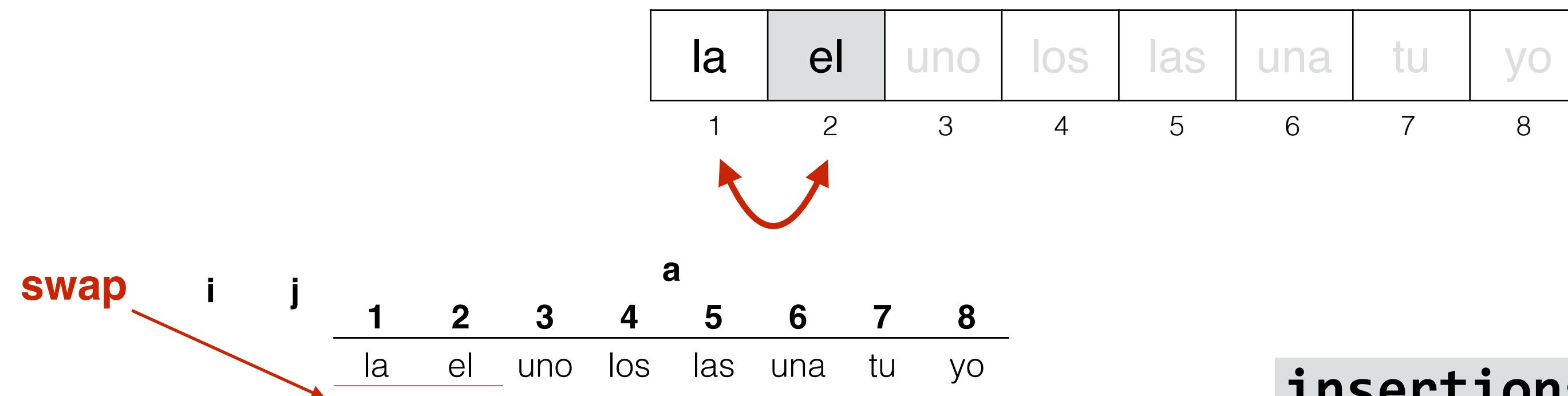
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

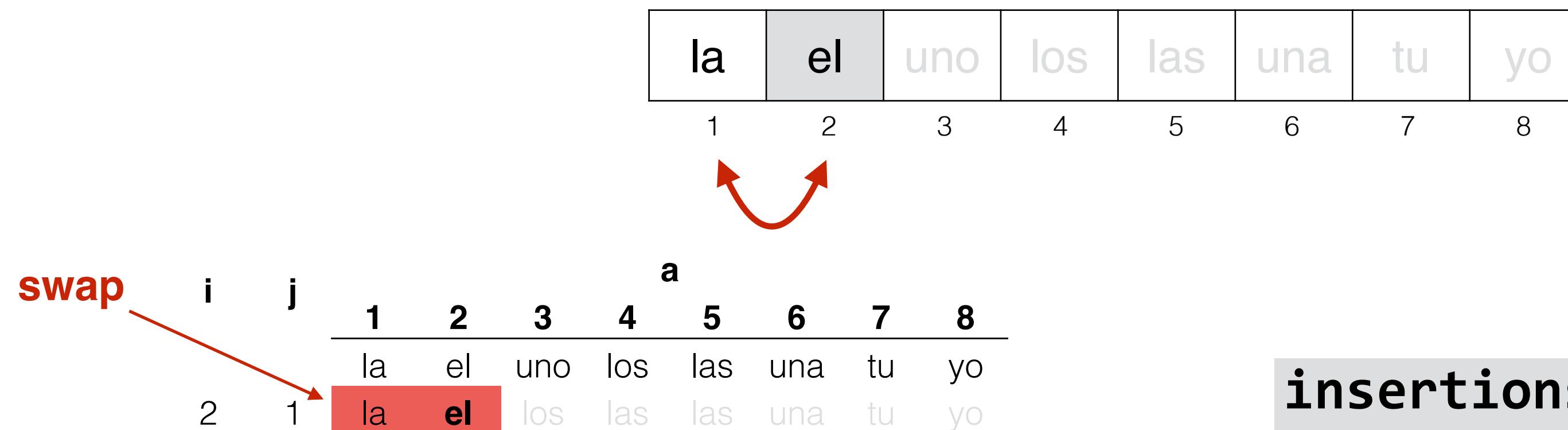
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

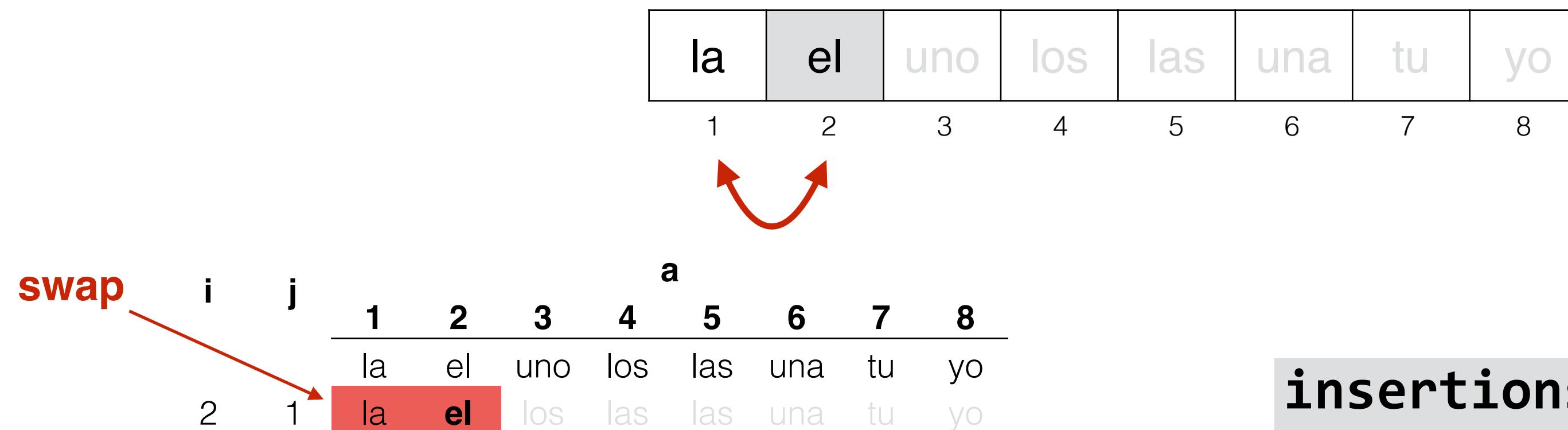
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

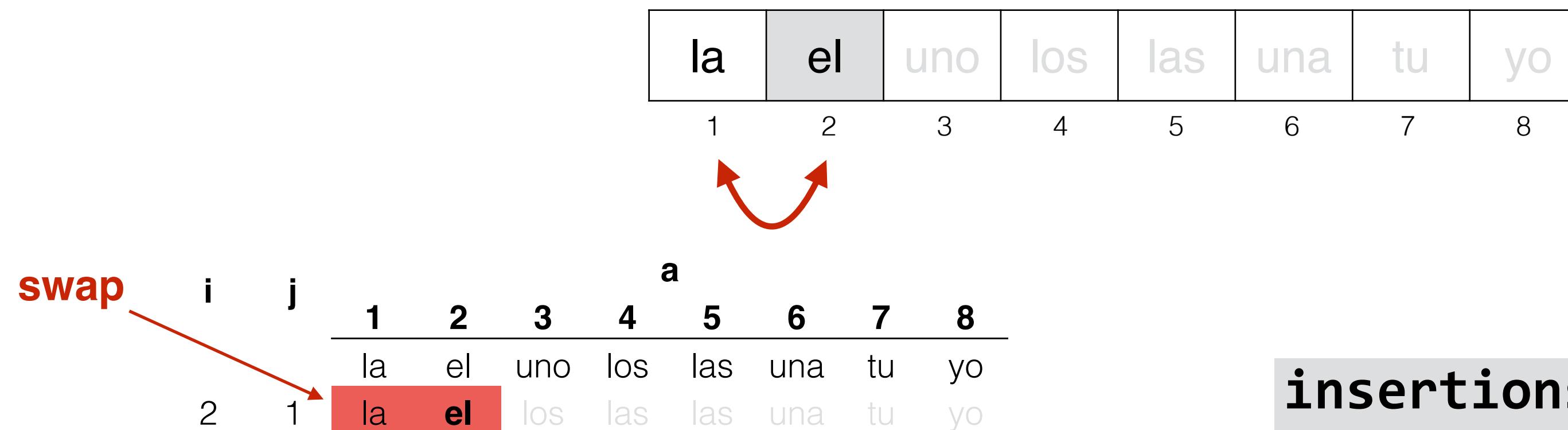
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

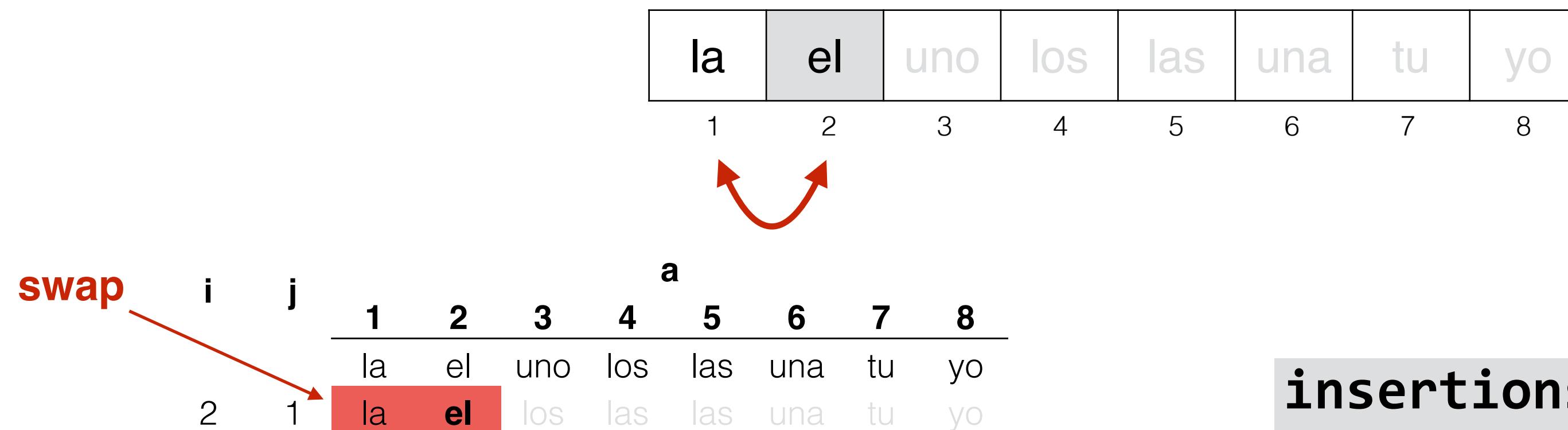
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

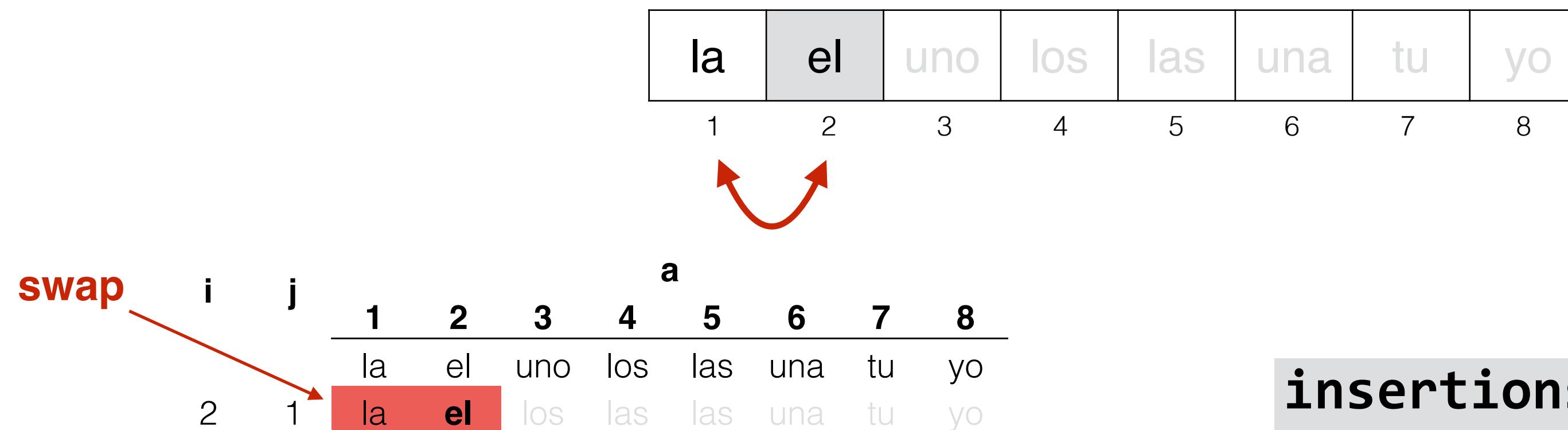
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

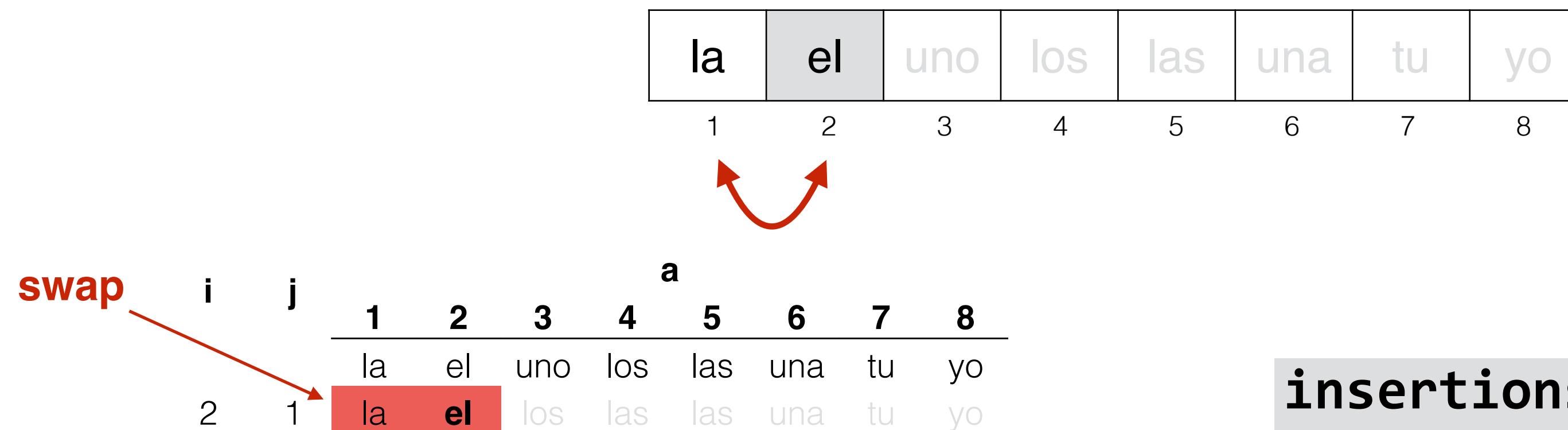
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

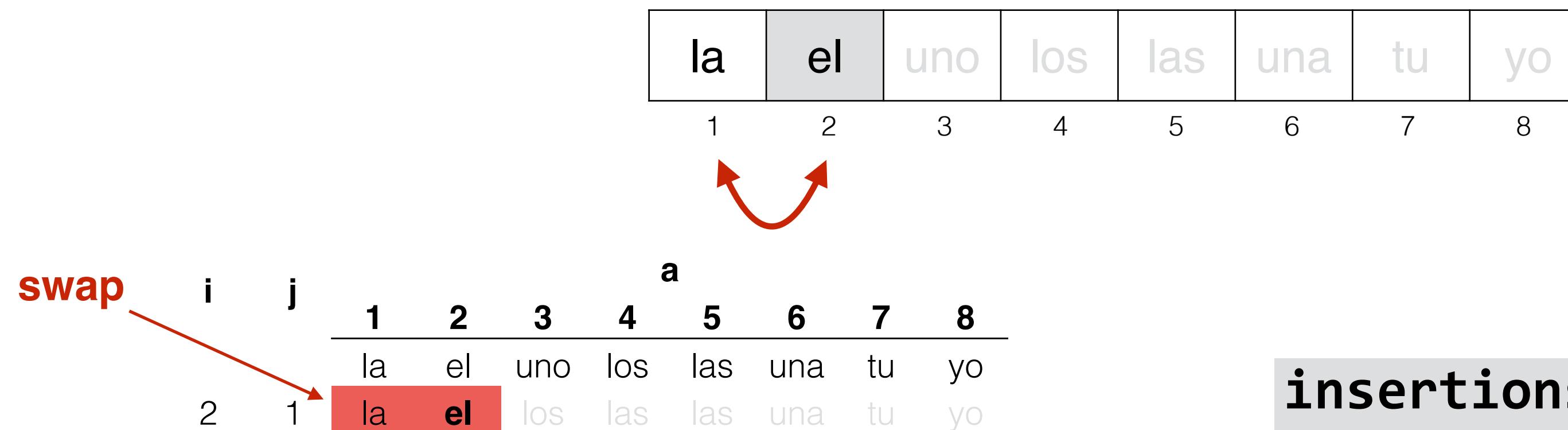
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

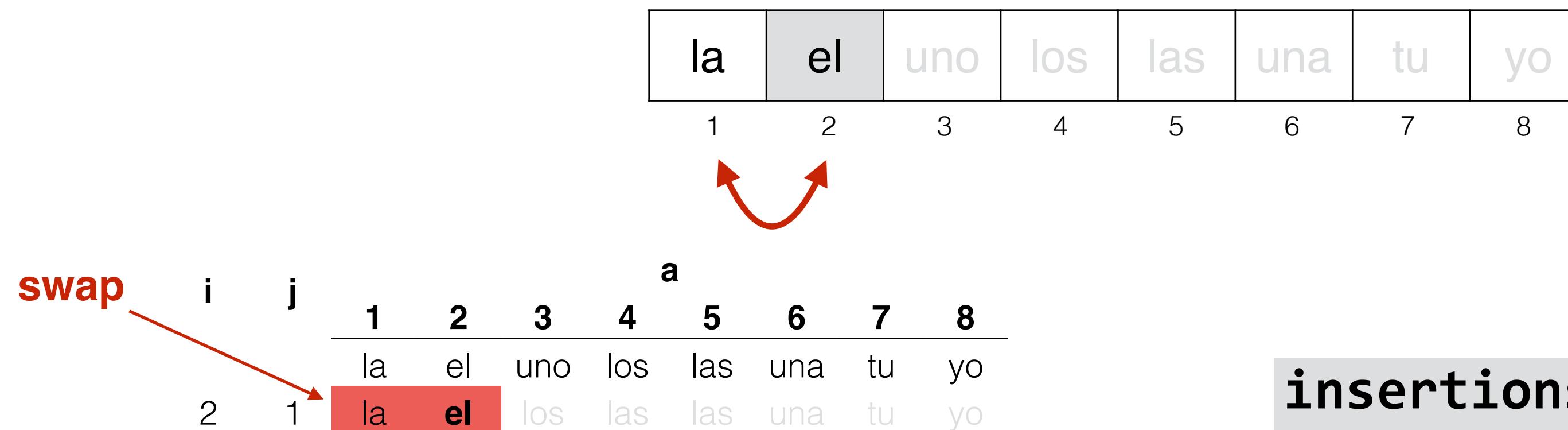
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

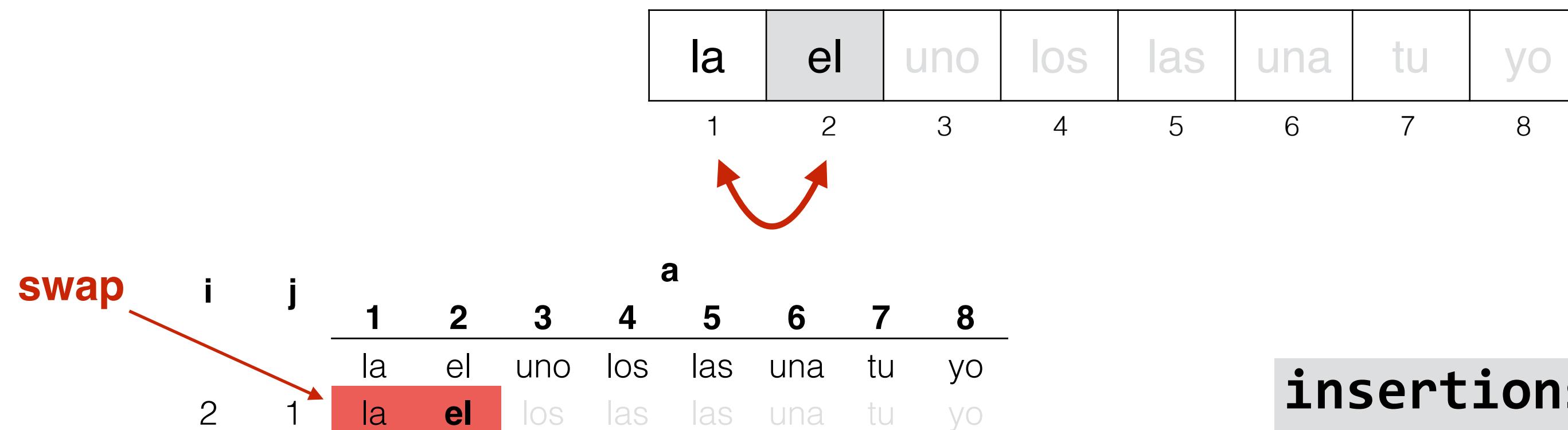
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

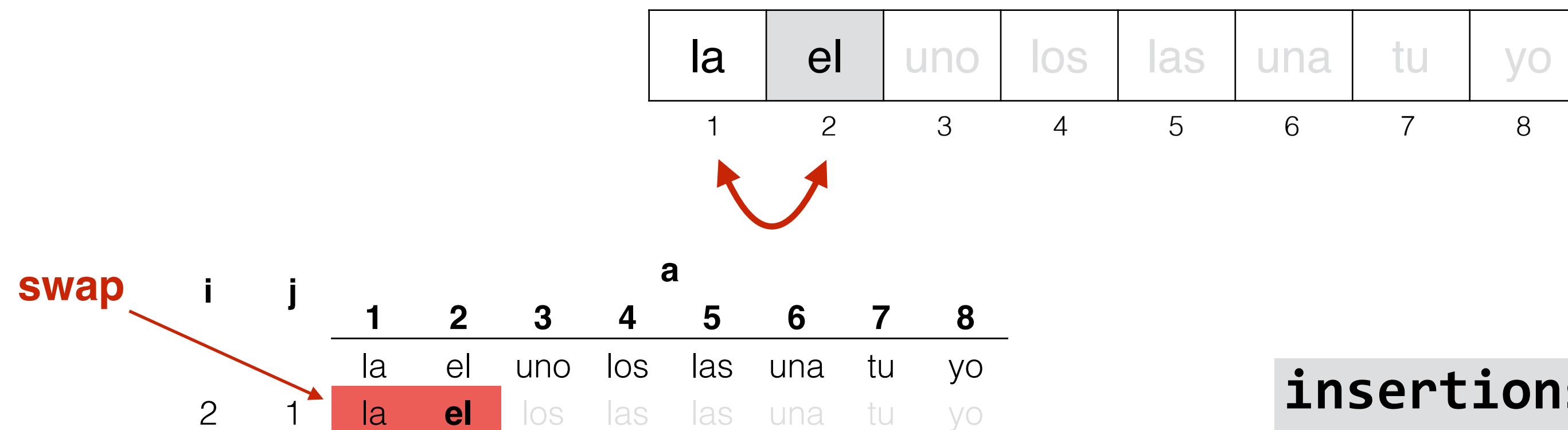
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

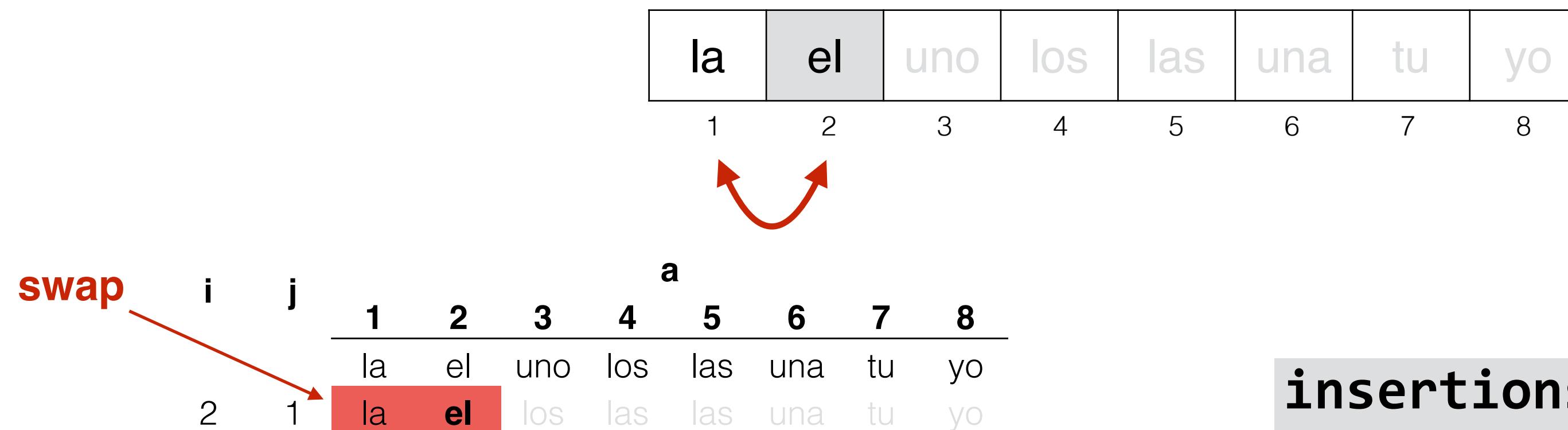
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

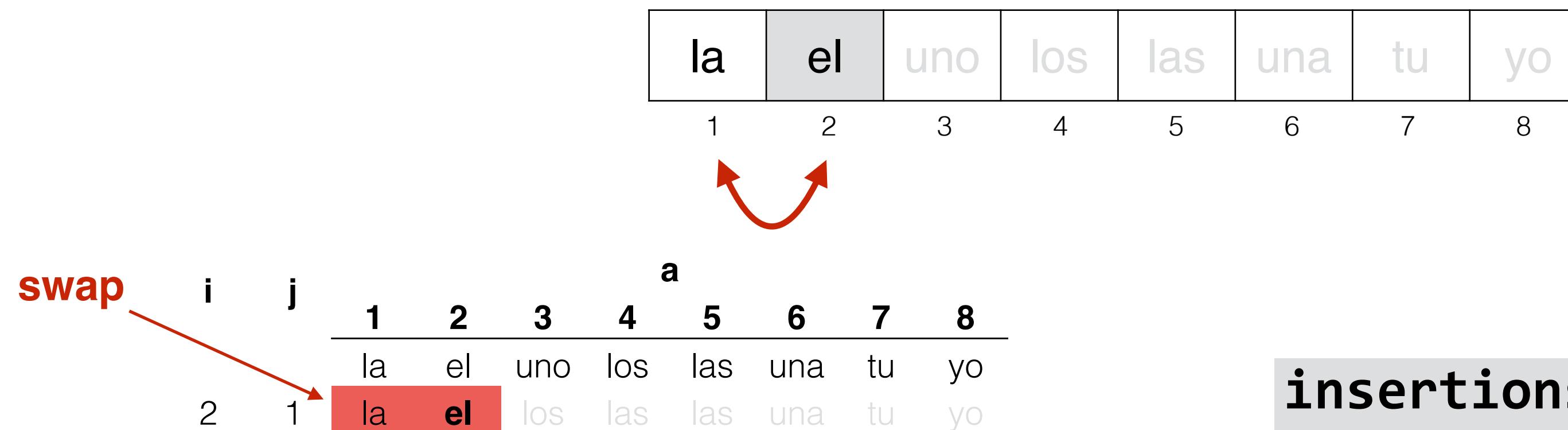
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

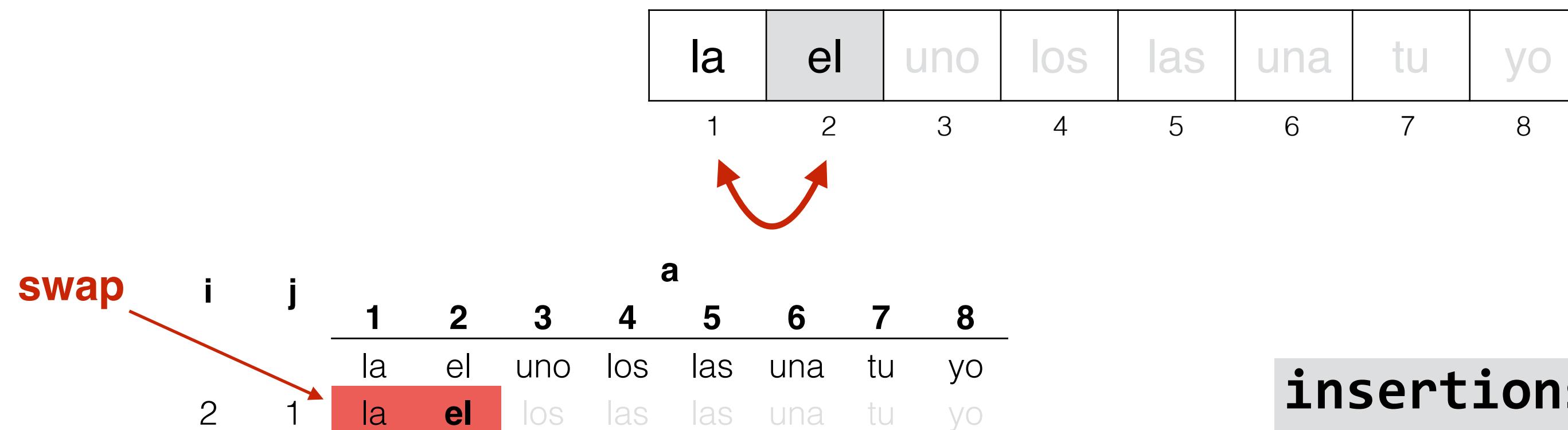
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

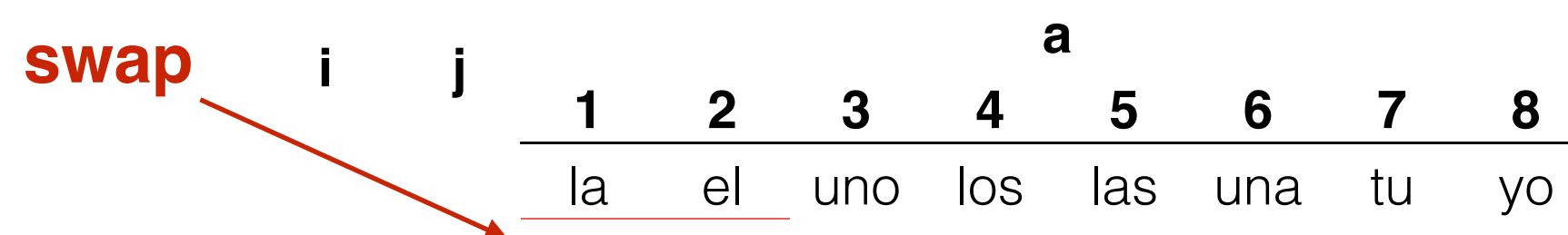
$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

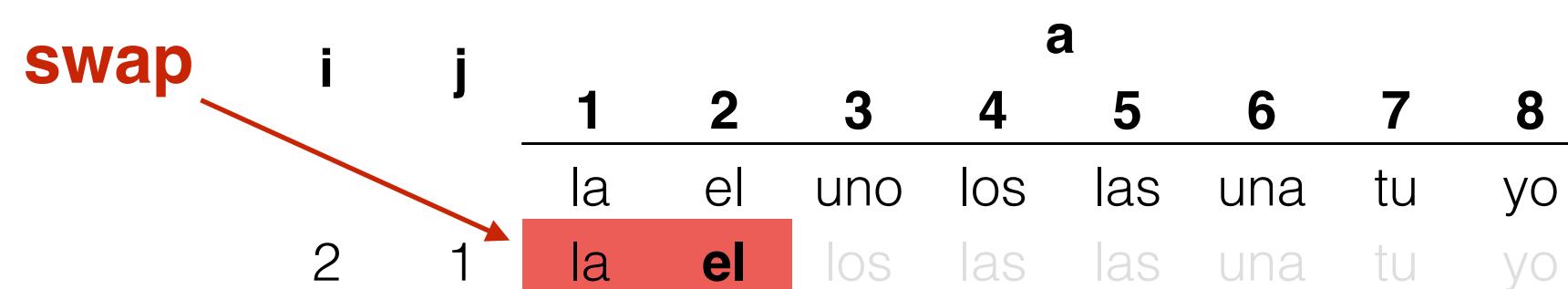
$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8

swap

	i	j	a	1	2	3	4	5	6	7	8
2	1	1	la	el	uno	los	las	una	tu	yo	
3	2	2	el	la	uno	las	las	una	tu	yo	

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8

swap

	i	j	a	1	2	3	4	5	6	7	8
2	1	1	la el uno los las una tu yo	la	el	uno	los	las	una	tu	yo
3	2	2	el la uno las las una tu yo	el	la	uno	las	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8

swap

		a							
		1	2	3	4	5	6	7	8
i	j	la	el	uno	los	las	una	tu	yo
		la	el	uno	los	las	una	tu	yo

2 1

3 2 el la uno las las una tu yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8

swap

	i	j	a	1	2	3	4	5	6	7	8
2	1	1	la el uno los las una tu yo	la	el	uno	los	las	una	tu	yo
3	2	2	el la uno las las una tu yo	el	la	uno	las	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8

swap

	i	j	a	1	2	3	4	5	6	7	8
2	1	1	la el uno los las una tu yo	la	el	uno	los	las	una	tu	yo
3	2	2	el la uno las las una tu yo	el	la	uno	las	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8

swap

	i	j	a	1	2	3	4	5	6	7	8
2	1	1	la	el	uno	los	las	una	tu	yo	
3	2	2	el	la	uno	las	las	una	tu	yo	

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8

swap

	i	j	a	1	2	3	4	5	6	7	8
2	1	1	la el uno los las una tu yo	la	el	uno	los	las	una	tu	yo
3	2	2	el la uno las las una tu yo	el	la	uno	las	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8

swap

	i	j	a	1	2	3	4	5	6	7	8
2	1	1	la el uno los las una tu yo	la	el	uno	los	las	una	tu	yo
3	2	2	el la uno las las una tu yo	el	la	uno	las	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8

swap

	i	j	a	1	2	3	4	5	6	7	8
2	1	1	la	el	uno	los	las	una	tu	yo	
3	2	2	el	la	uno	las	las	una	tu	yo	

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8

swap

	i	j	a	1	2	3	4	5	6	7	8
2	1	1	la el uno los las una tu yo	la	el	uno	los	las	una	tu	yo
3	2	2	el la uno las las una tu yo	el	la	uno	las	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8

swap

		a							
		1	2	3	4	5	6	7	8
i	j	la	el	uno	los	las	una	tu	yo
		la	el	uno	los	las	una	tu	yo

2 1

3 2 el la uno las las una tu yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	uno	los	las	una	tu	yo
1	2	3	4	5	6	7	8

swap

	i	j	a	1	2	3	4	5	6	7	8
2	1	1	la el uno los las una tu yo	la	el	uno	los	las	una	tu	yo
3	2	2	el la uno las las una tu yo	el	la	uno	las	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

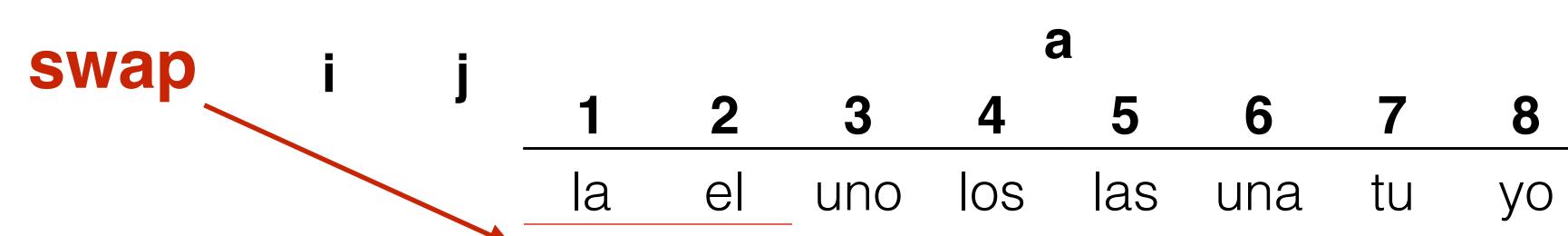
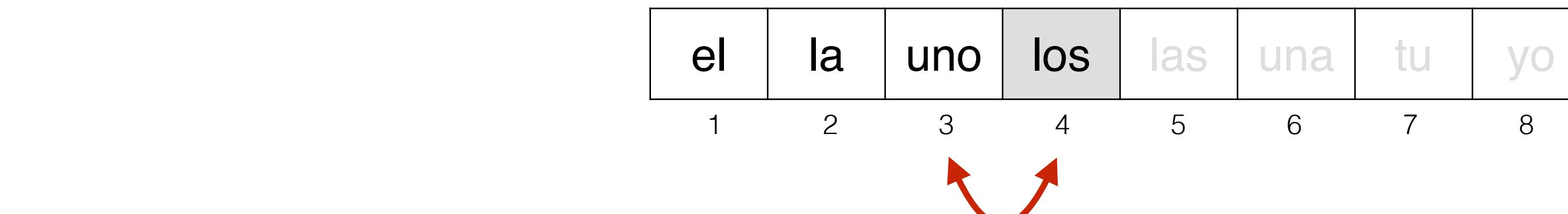
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

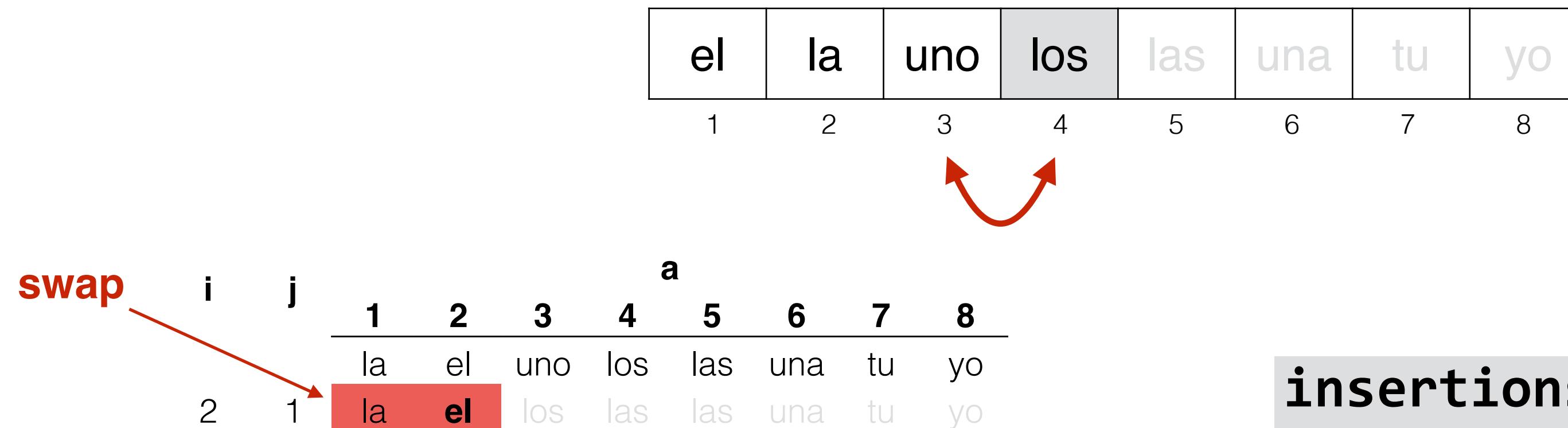
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

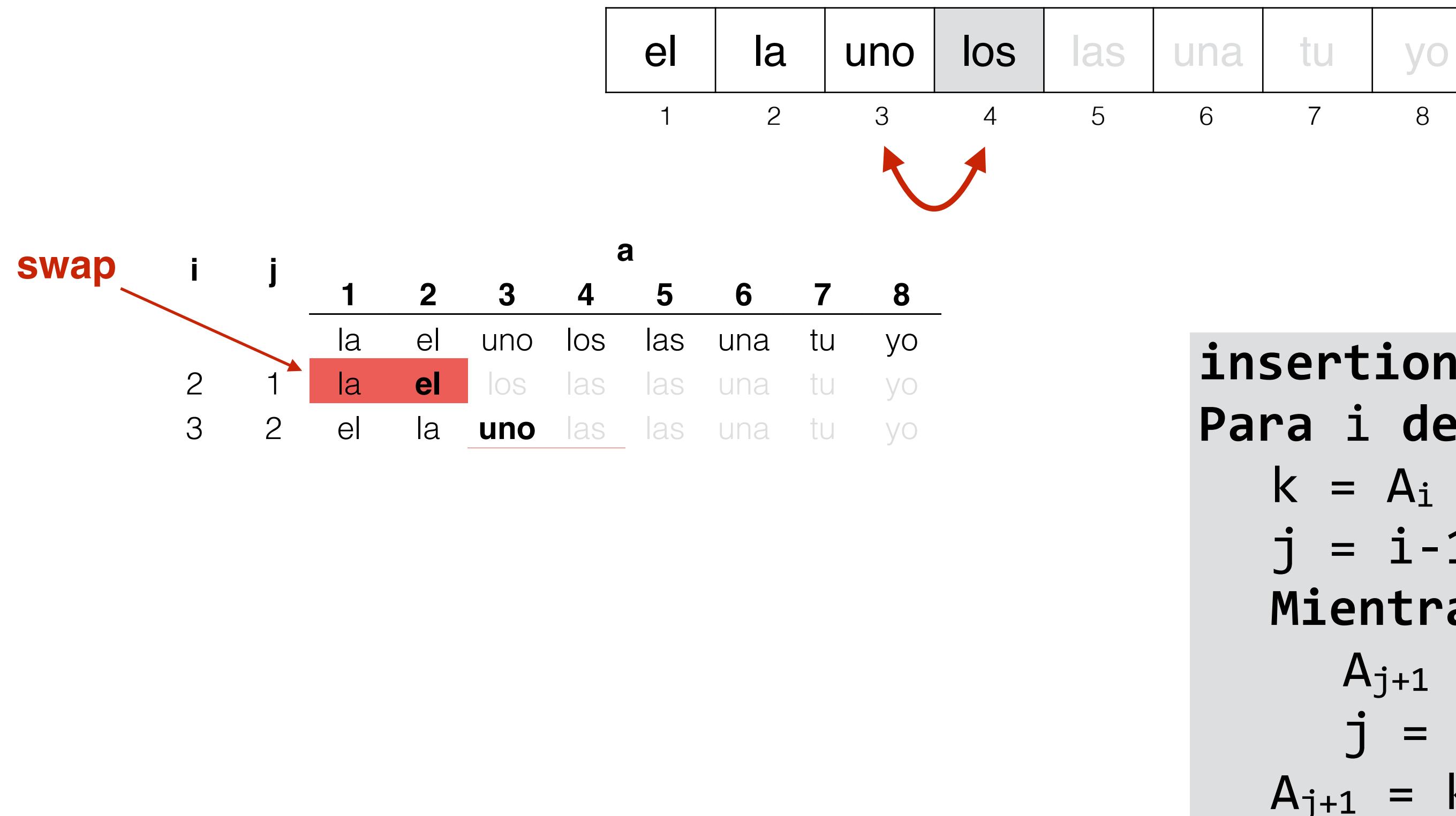
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	uno	los	las	una	tu	yo		
	1	2	3	4	5	6	7	8		
swap	i	j	a							
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	uno	los	las	una	tu	yo
	1	2	3	4	5	6	7	8

swap i j

	1	2	3	4	5	6	7	8
2	la	el	uno	los	las	una	tu	yo
3	el	la	uno	las	las	una	tu	yo
4	el	la	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	uno	los	las	una	tu	yo
	1	2	3	4	5	6	7	8

swap i j

	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	uno	los	las	una	tu	yo		
	1	2	3	4	5	6	7	8		
swap	i	j	a							
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	uno	los	las	una	tu	yo
	1	2	3	4	5	6	7	8

swap i j

	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	uno	los	las	una	tu	yo
	1	2	3	4	5	6	7	8

swap i j

	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	uno	los	las	una	tu	yo		
	1	2	3	4	5	6	7	8		
swap	i	j	a							
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	uno	los	las	una	tu	yo
	1	2	3	4	5	6	7	8

swap i j

	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	uno	los	las	una	tu	yo
	1	2	3	4	5	6	7	8

swap i j

	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	uno	los	las	una	tu	yo
	1	2	3	4	5	6	7	8

swap i j

	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	uno	los	las	una	tu	yo
	1	2	3	4	5	6	7	8

swap i j

	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

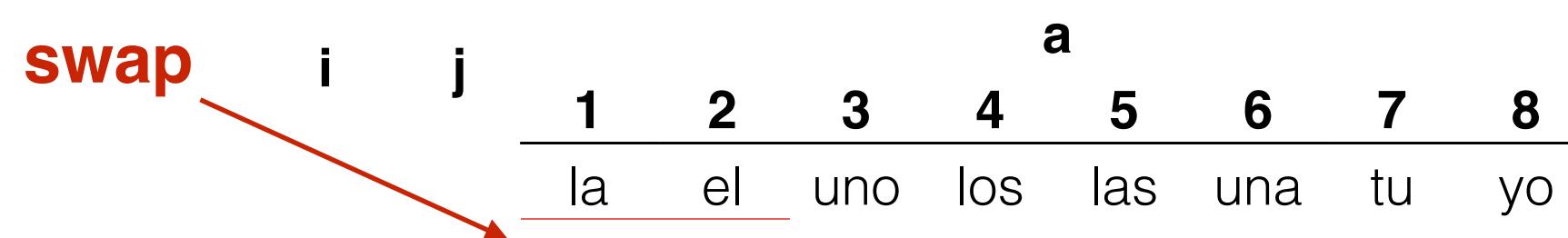
$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	los	uno	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

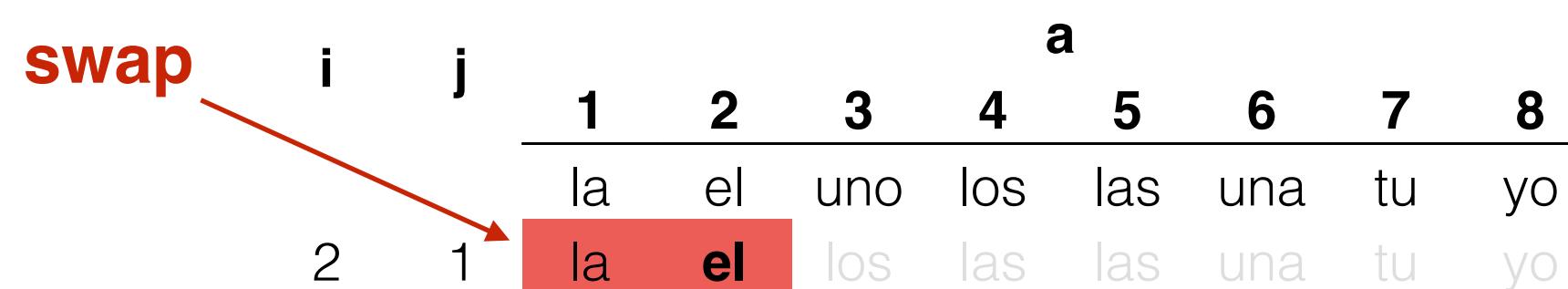
$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	los	uno	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	los	uno	las	una	tu	yo
1	2	3	4	5	6	7	8

swap

	i	j	a	1	2	3	4	5	6	7	8
2	1	1	la el	uno	los	las	una	tu	yo		
3	2	2	el la	uno	las	las	una	tu	yo		

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	los	uno	las	una	tu	yo
1	2	3	4	5	6	7	8

	i	j	a	1	2	3	4	5	6	7	8	
swap				la	el	uno	los	las	una	tu	yo	
2	1	1	la	el	el	uno	los	las	las	una	tu	yo
3	2	2	el	la	el	uno	las	las	una	tu	yo	
4	3	3	el	la	el	uno	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	los	uno	las	una	tu	yo
1	2	3	4	5	6	7	8

	i	j	a	1	2	3	4	5	6	7	8
				la	el	uno	los	las	una	tu	yo
2	1	1	swap	la	el	los	las	las	una	tu	yo
3	2	2		el	la	uno	las	las	una	tu	yo
4	3	3		el	la	uno	los	las	una	tu	yo
4	2	2		el	la	los	uno	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	los	uno	las	una	tu	yo
1	2	3	4	5	6	7	8

	i	j	a	1	2	3	4	5	6	7	8
				la	el	uno	los	las	una	tu	yo
2	1	1	swap	la	el	los	las	las	una	tu	yo
3	2	2		el	la	uno	las	las	una	tu	yo
4	3	3		el	la	uno	los	las	una	tu	yo
4	2	2		el	la	los	uno	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	los	uno	las	una	tu	yo
1	2	3	4	5	6	7	8

	i	j	a	1	2	3	4	5	6	7	8
				la	el	uno	los	las	una	tu	yo
2	1	1	swap	la	el	los	las	las	una	tu	yo
3	2	2		el	la	uno	las	las	una	tu	yo
4	3	3		el	la	uno	los	las	una	tu	yo
4	2	2		el	la	los	uno	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	los	uno	las	una	tu	yo
1	2	3	4	5	6	7	8

	i	j	a	1	2	3	4	5	6	7	8
				la	el	uno	los	las	una	tu	yo
2	1	1	swap	la	el	los	las	las	una	tu	yo
3	2	2		el	la	uno	las	las	una	tu	yo
4	3	3		el	la	uno	los	las	una	tu	yo
4	2	2		el	la	los	uno	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	los	uno	las	una	tu	yo
1	2	3	4	5	6	7	8

	i	j	a	1	2	3	4	5	6	7	8
				la	el	uno	los	las	una	tu	yo
2	1	1	swap	la	el	los	las	las	una	tu	yo
3	2	2		el	la	uno	las	las	una	tu	yo
4	3	3		el	la	uno	los	las	una	tu	yo
4	2	2		el	la	los	uno	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	los	uno	las	una	tu	yo
1	2	3	4	5	6	7	8

	i	j	a	1	2	3	4	5	6	7	8
				la	el	uno	los	las	una	tu	yo
2	1	1	swap	la	el	los	las	las	una	tu	yo
3	2	2		el	la	uno	las	las	una	tu	yo
4	3	3		el	la	uno	los	las	una	tu	yo
4	2	2		el	la	los	uno	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	los	uno	las	una	tu	yo
1	2	3	4	5	6	7	8

	i	j	a	1	2	3	4	5	6	7	8
				la	el	uno	los	las	una	tu	yo
2	1	1	swap	la	el	los	las	las	una	tu	yo
3	2	2		el	la	uno	las	las	una	tu	yo
4	3	3		el	la	uno	los	las	una	tu	yo
4	2	2		el	la	los	uno	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	los	uno	las	una	tu	yo
1	2	3	4	5	6	7	8

swap	i	j	a							
			1	2	3	4	5	6	7	8
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo
	4	2	el	la	los	uno	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	los	uno	las	una	tu	yo
1	2	3	4	5	6	7	8

	i	j	a	1	2	3	4	5	6	7	8
				la	el	uno	los	las	una	tu	yo
2	1	1	swap	la	el	los	las	las	una	tu	yo
3	2	2		el	la	uno	las	las	una	tu	yo
4	3	3		el	la	uno	los	las	una	tu	yo
4	2	2		el	la	los	uno	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	los	uno	las	una	tu	yo
1	2	3	4	5	6	7	8

	i	j	a	1	2	3	4	5	6	7	8
				la	el	uno	los	las	una	tu	yo
2	1	1	swap	la	el	los	las	las	una	tu	yo
3	2	2		el	la	uno	las	las	una	tu	yo
4	3	3		el	la	uno	los	las	una	tu	yo
4	2	2		el	la	los	uno	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

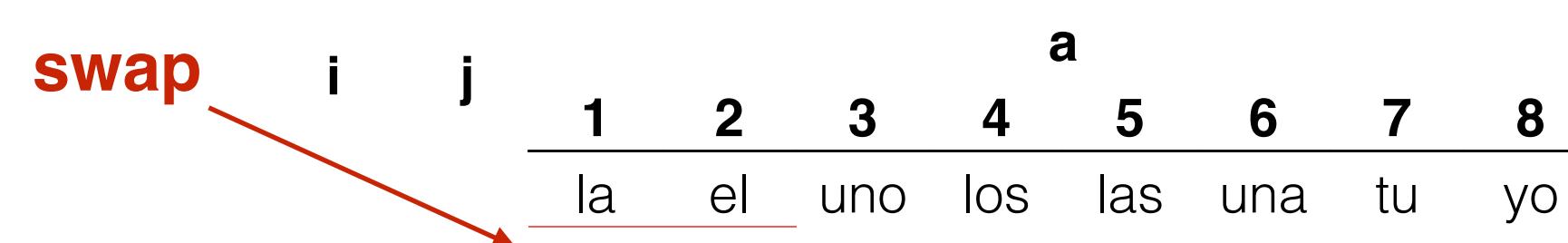
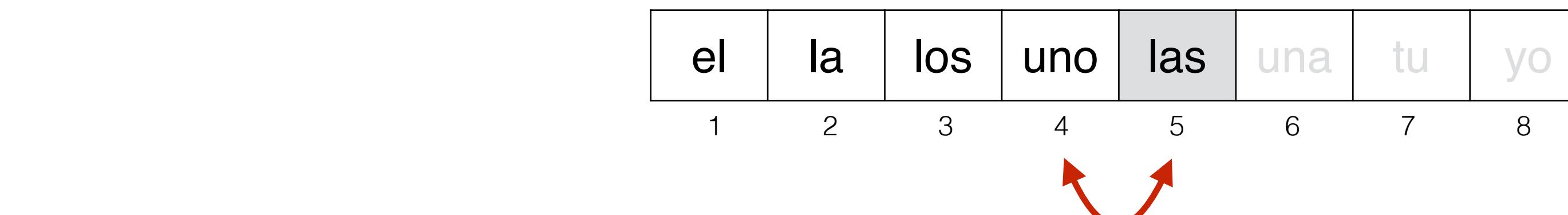
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

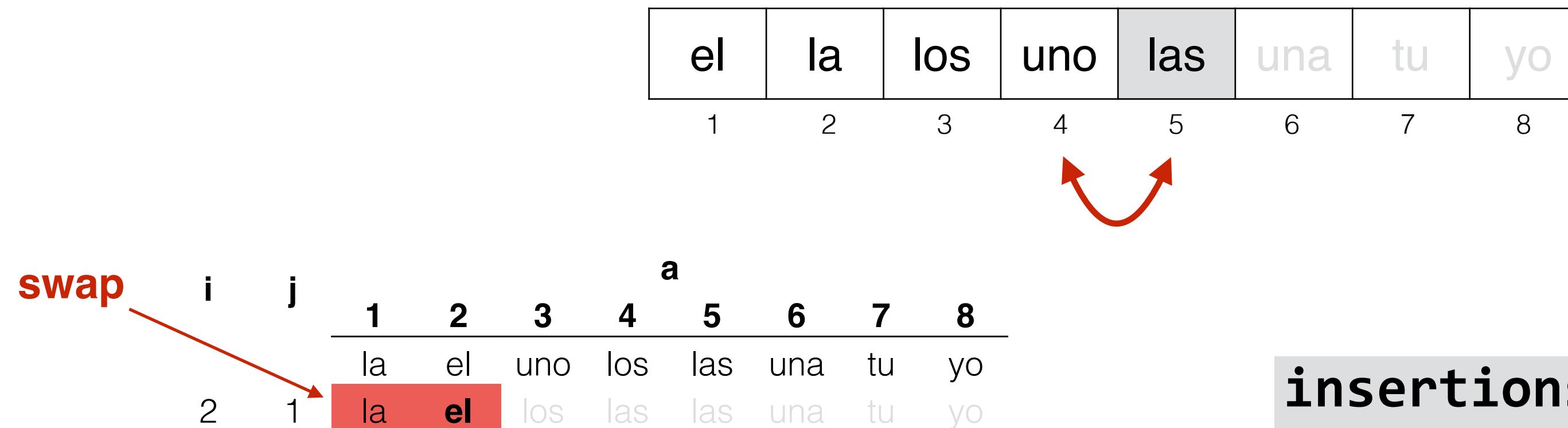
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

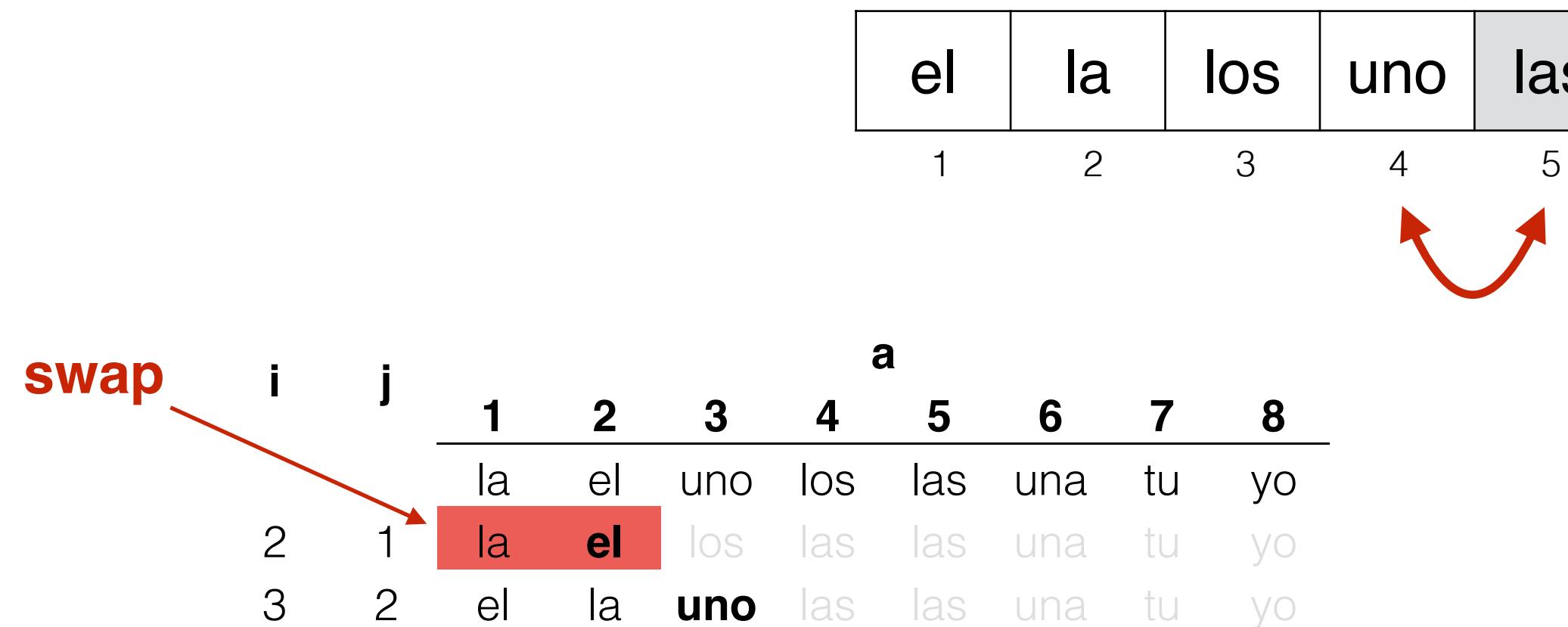
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

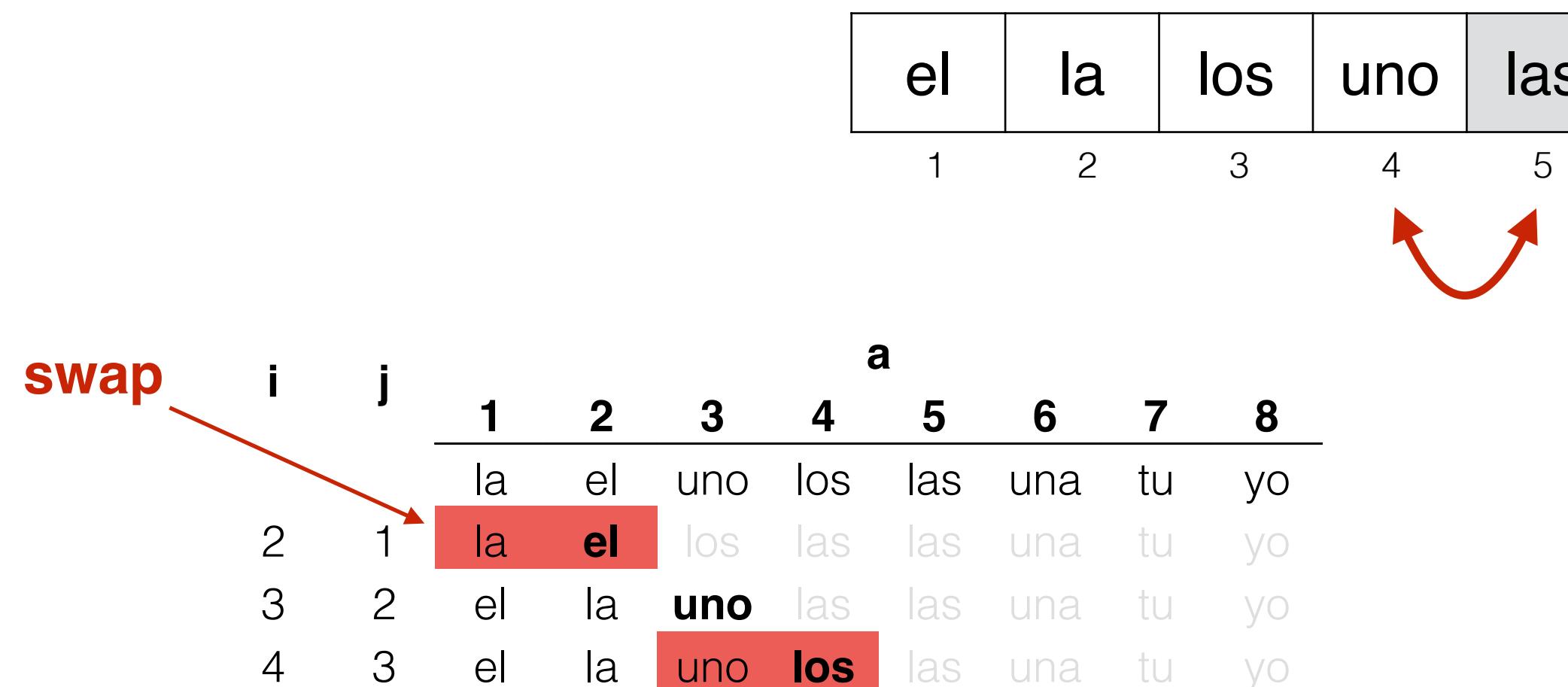
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo

i j



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

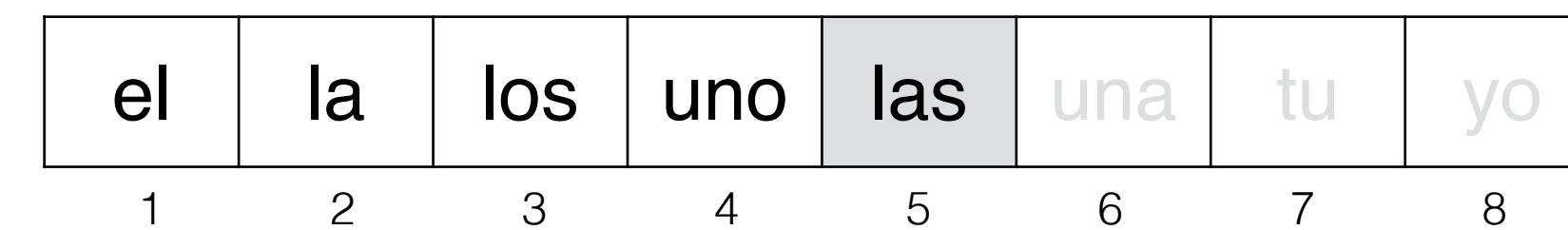
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

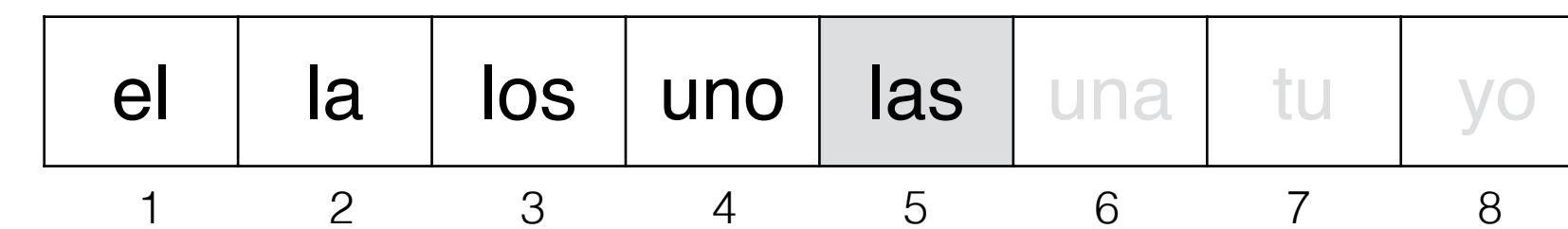
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

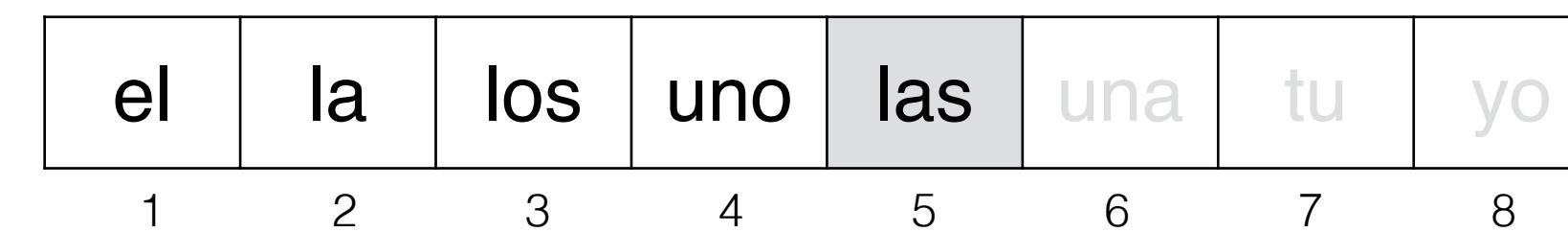
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

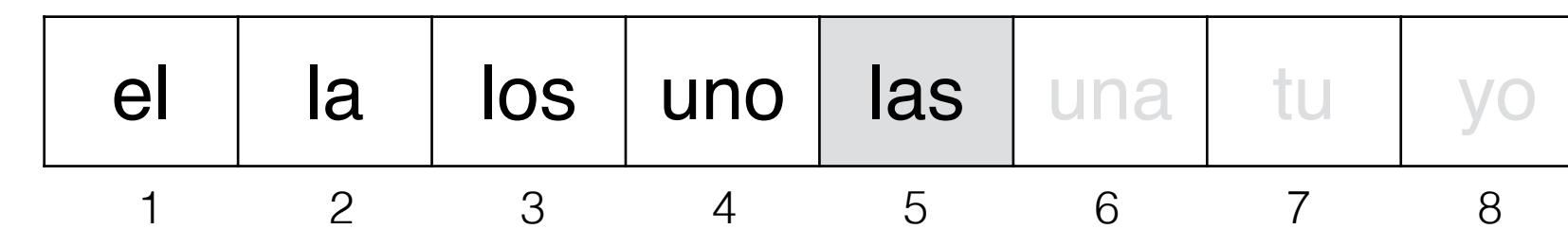
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

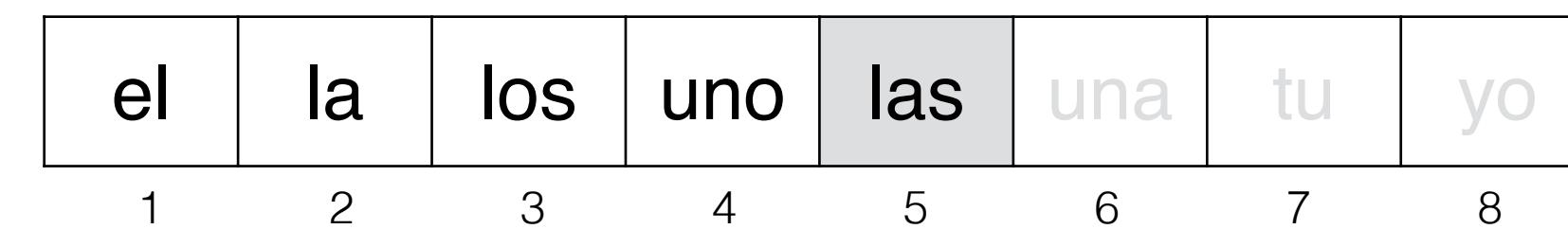
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo

el	la	los	uno	las	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

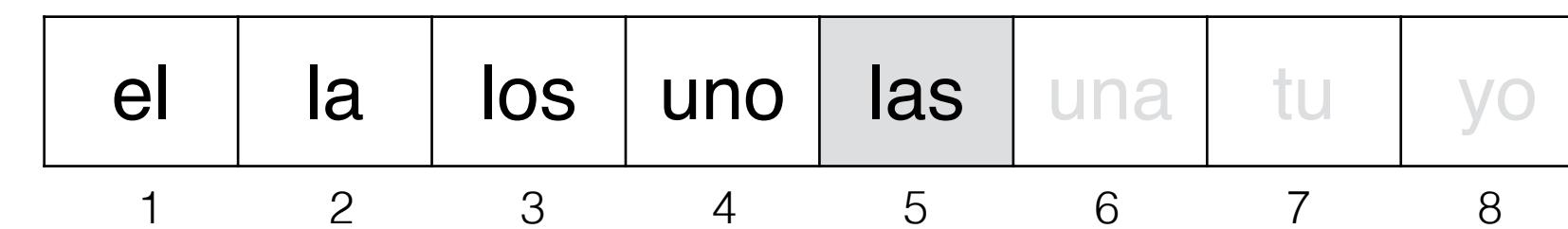
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

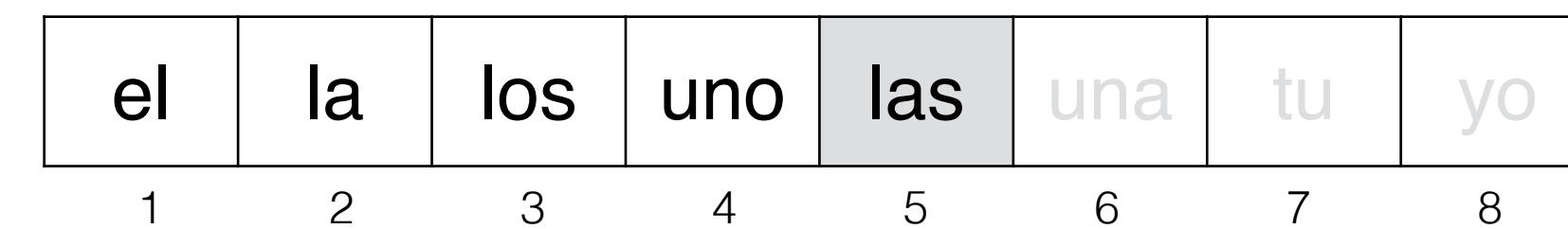
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

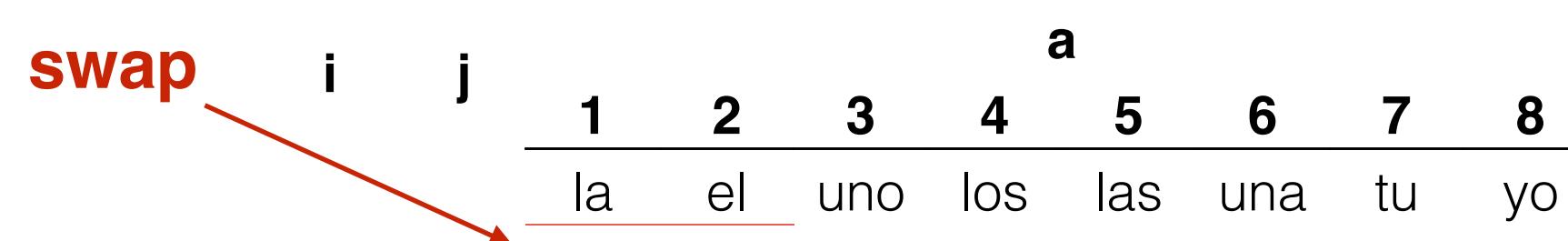
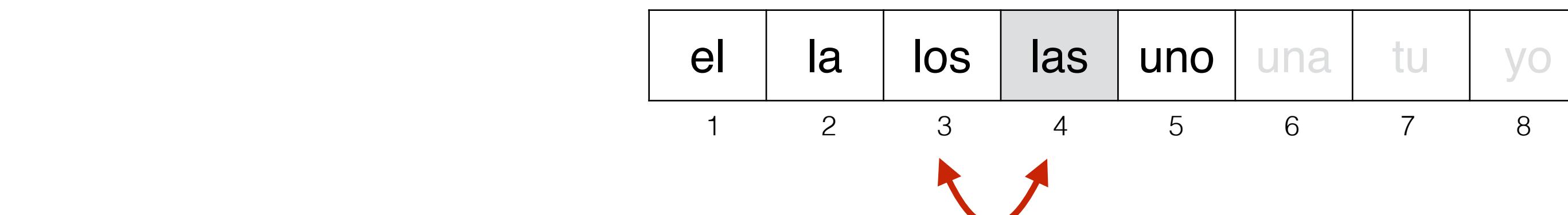
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

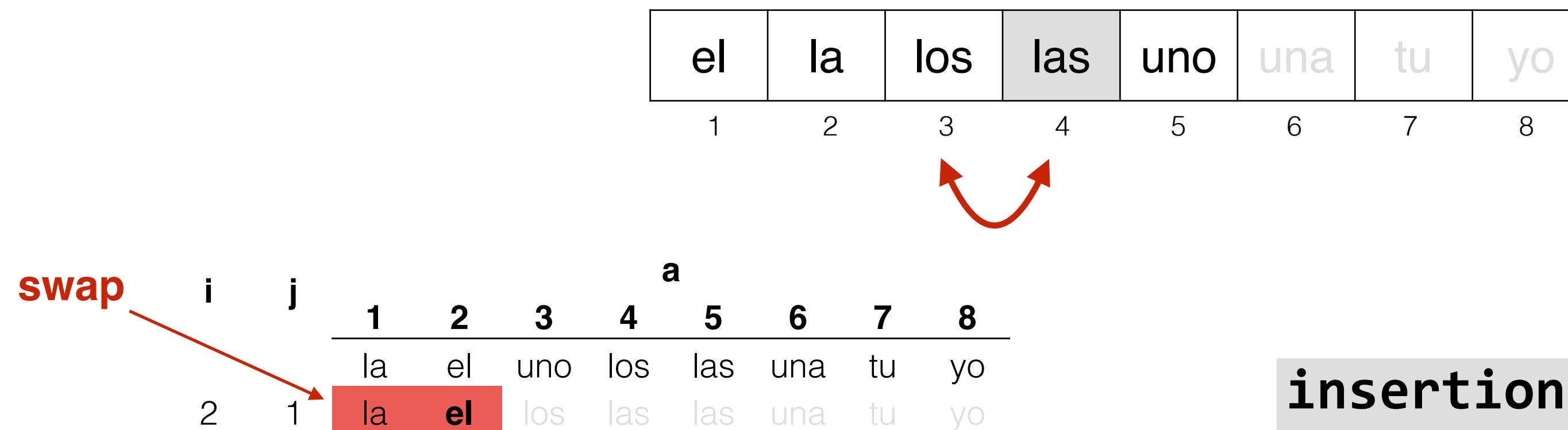
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

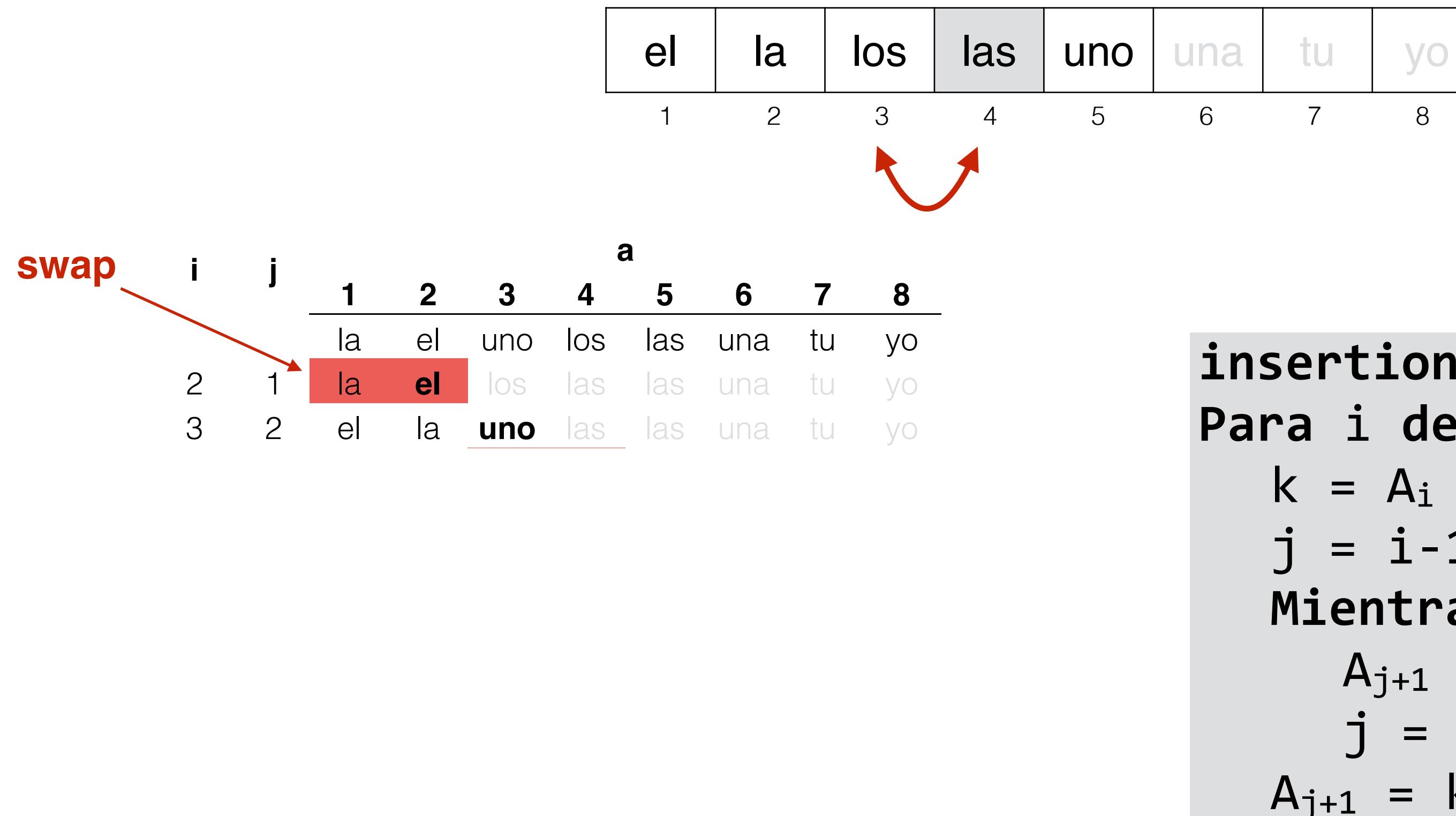
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo

i j

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo

i j

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

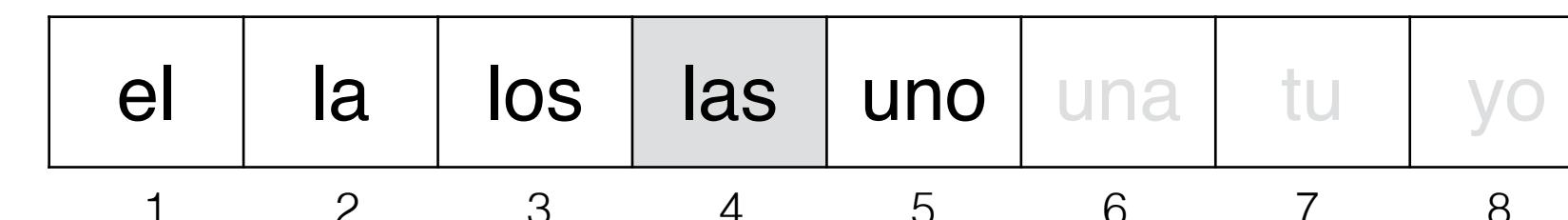
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a									
	1	2	3	4	5	6	7	8		
2	1	la	el	uno	los	las	una	tu	yo	
3	2	el	la	uno	las	las	una	tu	yo	
4	3	el	la	uno	los	las	una	tu	yo	
4	2	el	la	los	uno	las	una	tu	yo	
5	4	el	la	los	uno	los	una	tu	yo	
5	3	la	la	los	los	las	uno	una	tu	yo



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

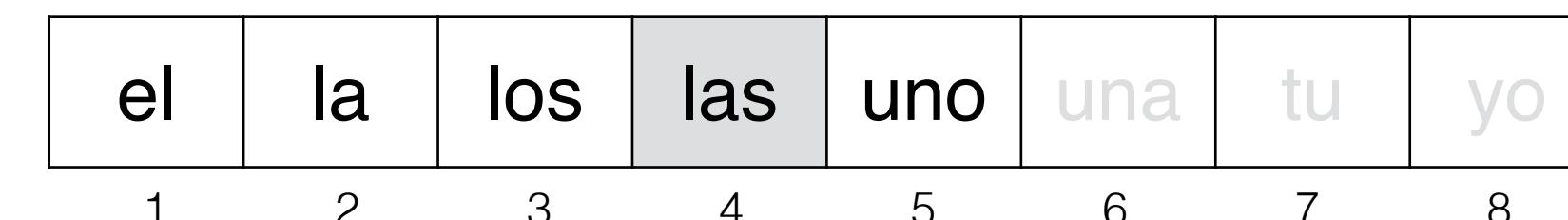
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a									
	1	2	3	4	5	6	7	8		
2	1	la	el	uno	los	las	una	tu	yo	
3	2	el	la	uno	las	las	una	tu	yo	
4	3	el	la	uno	los	las	una	tu	yo	
4	2	el	la	los	uno	las	una	tu	yo	
5	4	el	la	los	uno	los	una	tu	yo	
5	3	la	la	los	los	las	uno	una	tu	yo



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

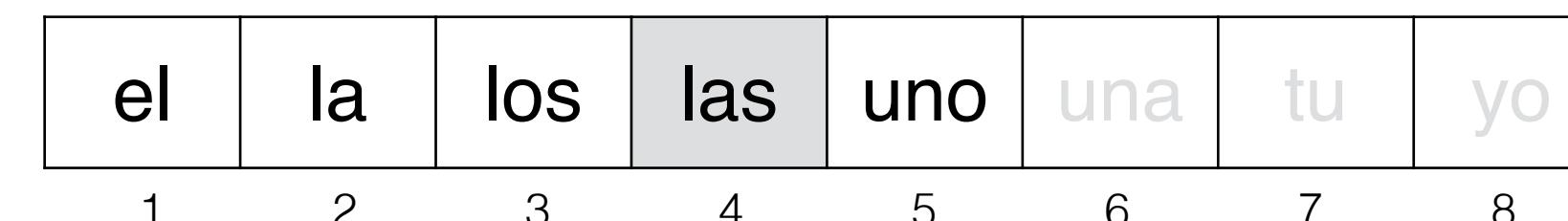
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a									
	1	2	3	4	5	6	7	8		
2	1	la	el	uno	los	las	una	tu	yo	
3	2	el	la	uno	las	las	una	tu	yo	
4	3	el	la	uno	los	las	una	tu	yo	
4	2	el	la	los	uno	las	una	tu	yo	
5	4	el	la	los	uno	los	una	tu	yo	
5	3	la	la	los	los	las	uno	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

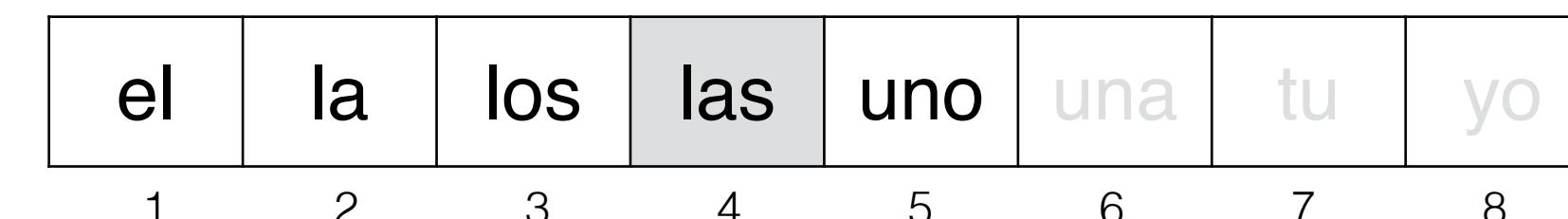
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a									
	1	2	3	4	5	6	7	8		
2	1	la	el	uno	los	las	una	tu	yo	
3	2	el	la	uno	las	las	una	tu	yo	
4	3	el	la	uno	los	las	una	tu	yo	
4	2	el	la	los	uno	las	una	tu	yo	
5	4	el	la	los	uno	los	una	tu	yo	
5	3	la	la	los	los	las	uno	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

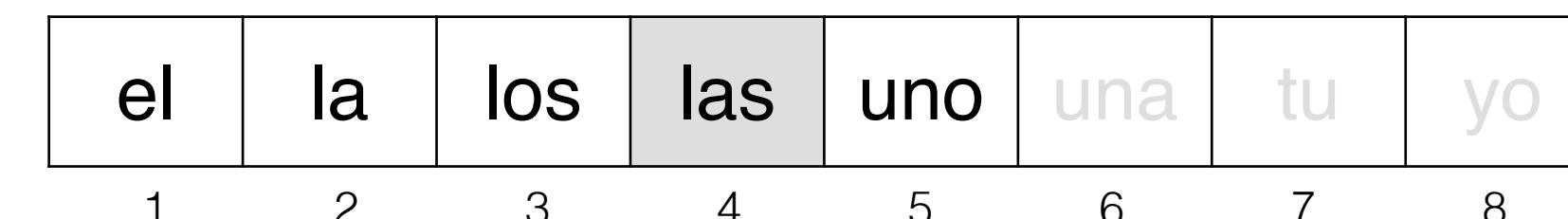
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a									
	1	2	3	4	5	6	7	8		
2	1	la	el	uno	los	las	una	tu	yo	
3	2	el	la	uno	las	las	una	tu	yo	
4	3	el	la	uno	los	las	una	tu	yo	
4	2	el	la	los	uno	las	una	tu	yo	
5	4	el	la	los	uno	los	una	tu	yo	
5	3	la	la	los	los	las	uno	una	tu	yo



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

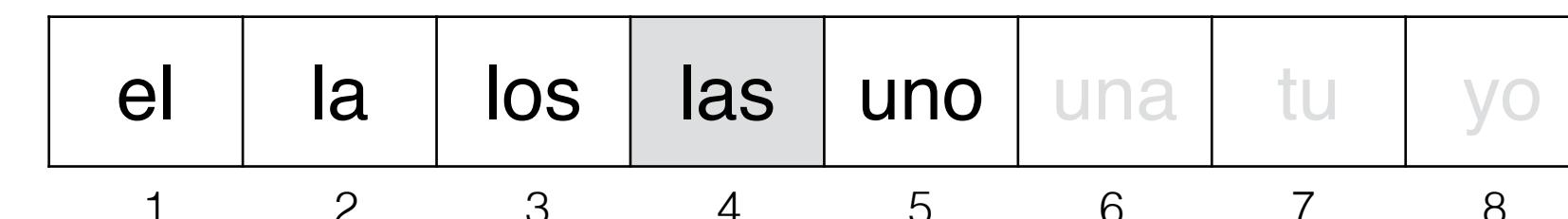
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a									
	1	2	3	4	5	6	7	8		
2	1	la	el	uno	los	las	una	tu	yo	
3	2	el	la	uno	las	las	una	tu	yo	
4	3	el	la	uno	los	las	una	tu	yo	
4	2	el	la	los	uno	las	una	tu	yo	
5	4	el	la	los	uno	los	una	tu	yo	
5	3	la	la	los	los	las	uno	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

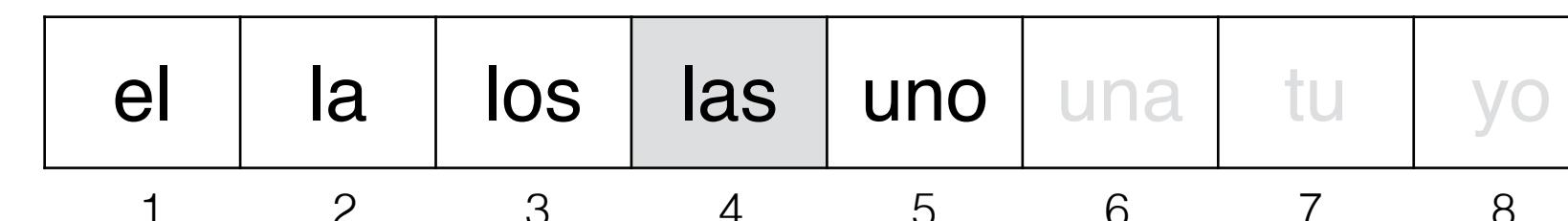
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a									
	1	2	3	4	5	6	7	8		
2	1	la	el	uno	los	las	una	tu	yo	
3	2	el	la	uno	las	las	una	tu	yo	
4	3	el	la	uno	los	las	una	tu	yo	
4	2	el	la	los	uno	las	una	tu	yo	
5	4	el	la	los	uno	los	una	tu	yo	
5	3	la	la	los	los	las	uno	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

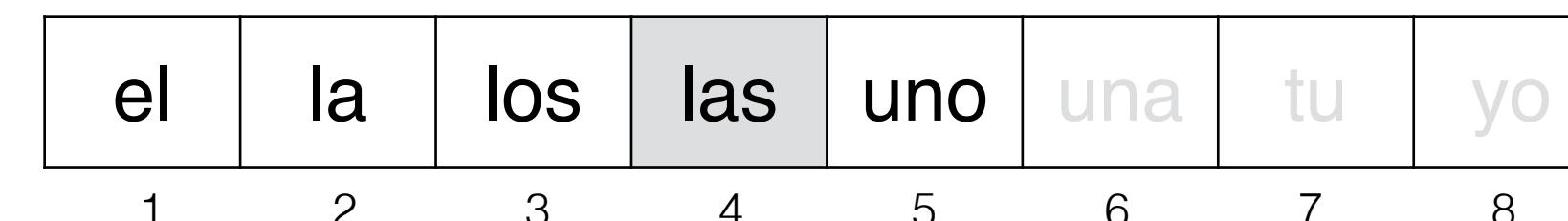
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a									
	1	2	3	4	5	6	7	8		
2	1	la	el	uno	los	las	una	tu	yo	
3	2	el	la	uno	las	las	una	tu	yo	
4	3	el	la	uno	los	las	una	tu	yo	
4	2	el	la	los	uno	las	una	tu	yo	
5	4	el	la	los	uno	los	una	tu	yo	
5	3	la	la	los	los	las	uno	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

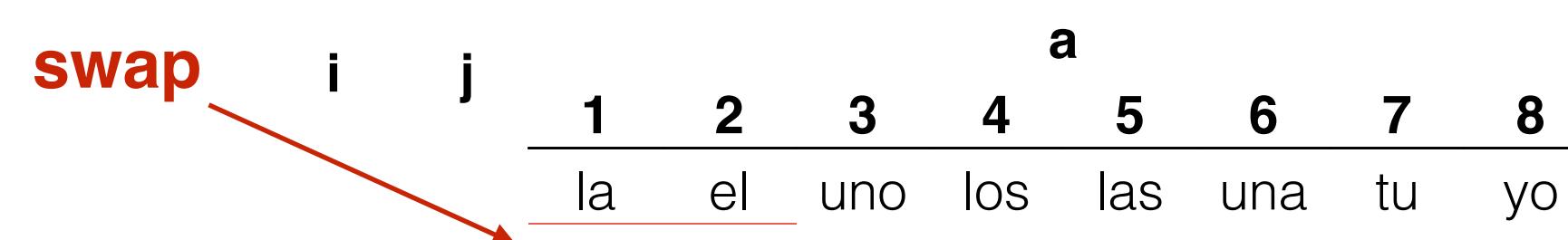
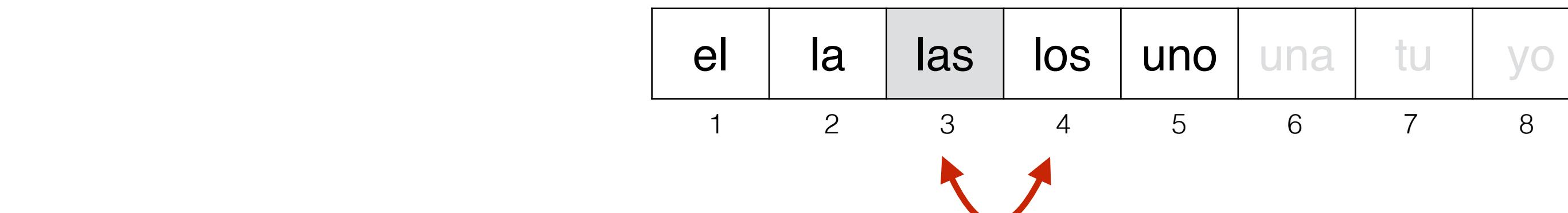
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

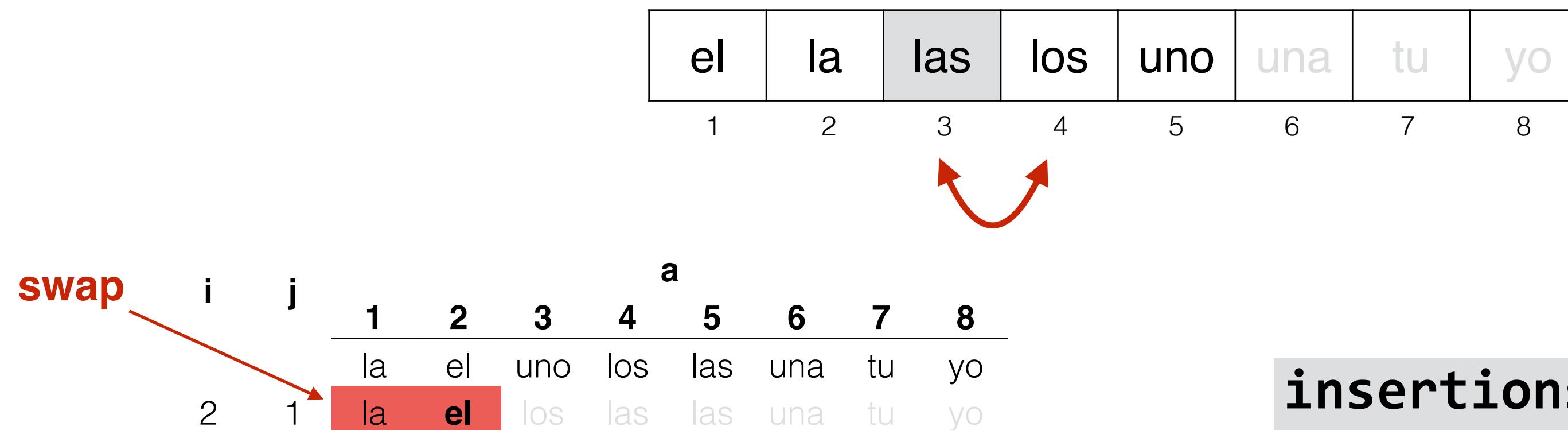
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

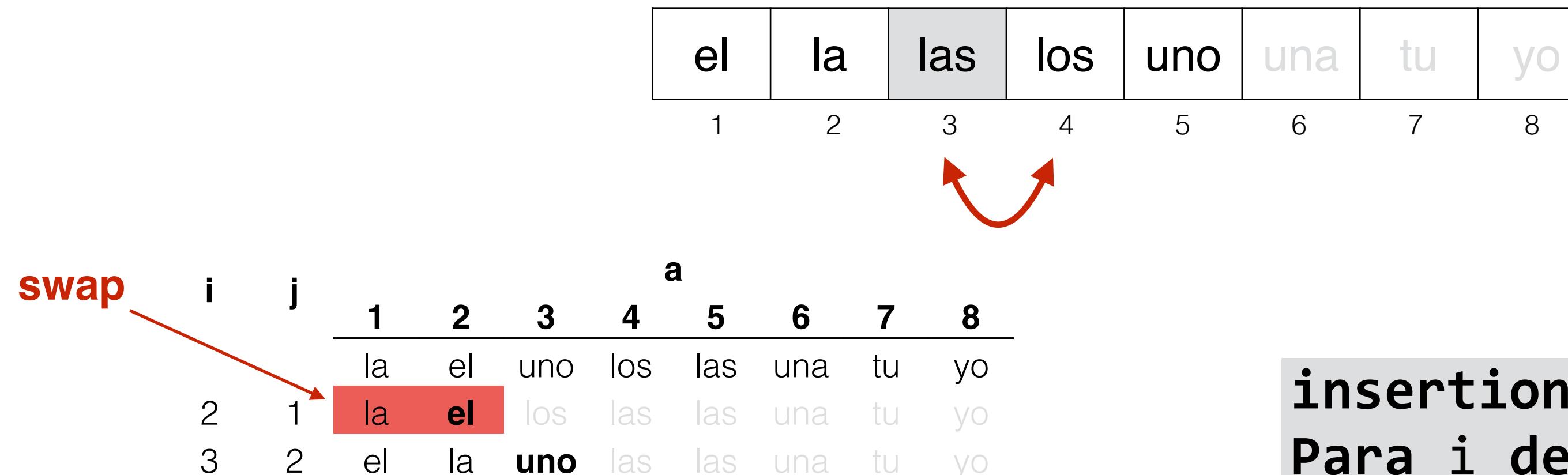
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	uno	una	tu	yo
1	2	3	4	5	6	7	8

swap

i j

	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

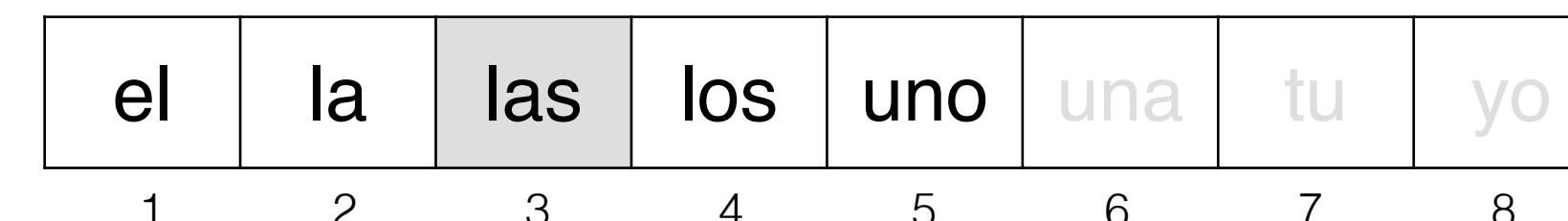
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	i	j	a	1	2	3	4	5	6	7	8	
	2	1	la el	la	el	uno	los	las	las	una	tu	yo
	3	2	el la	el	la	uno	los	las	las	una	tu	yo
	4	3	el la	el	la	uno	los	las	una	tu	yo	
	4	2	el la	el	la	los	uno	las	una	tu	yo	
	5	4	el la	el	la	los	uno	los	una	tu	yo	



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a							
	1	2	3	4	5	6	7	8
1	la	el	uno	los	las	una	tu	yo
2	1	la	el	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu
4	3	el	la	uno	los	las	una	tu
4	2	el	la	los	uno	las	una	tu
5	4	el	la	los	uno	las	una	tu
5	3	la	la	los	los	una	tu	yo

el	la	las	los	uno	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo

el	la	las	los	uno	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo

el	la	las	los	uno	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo

el	la	las	los	uno	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

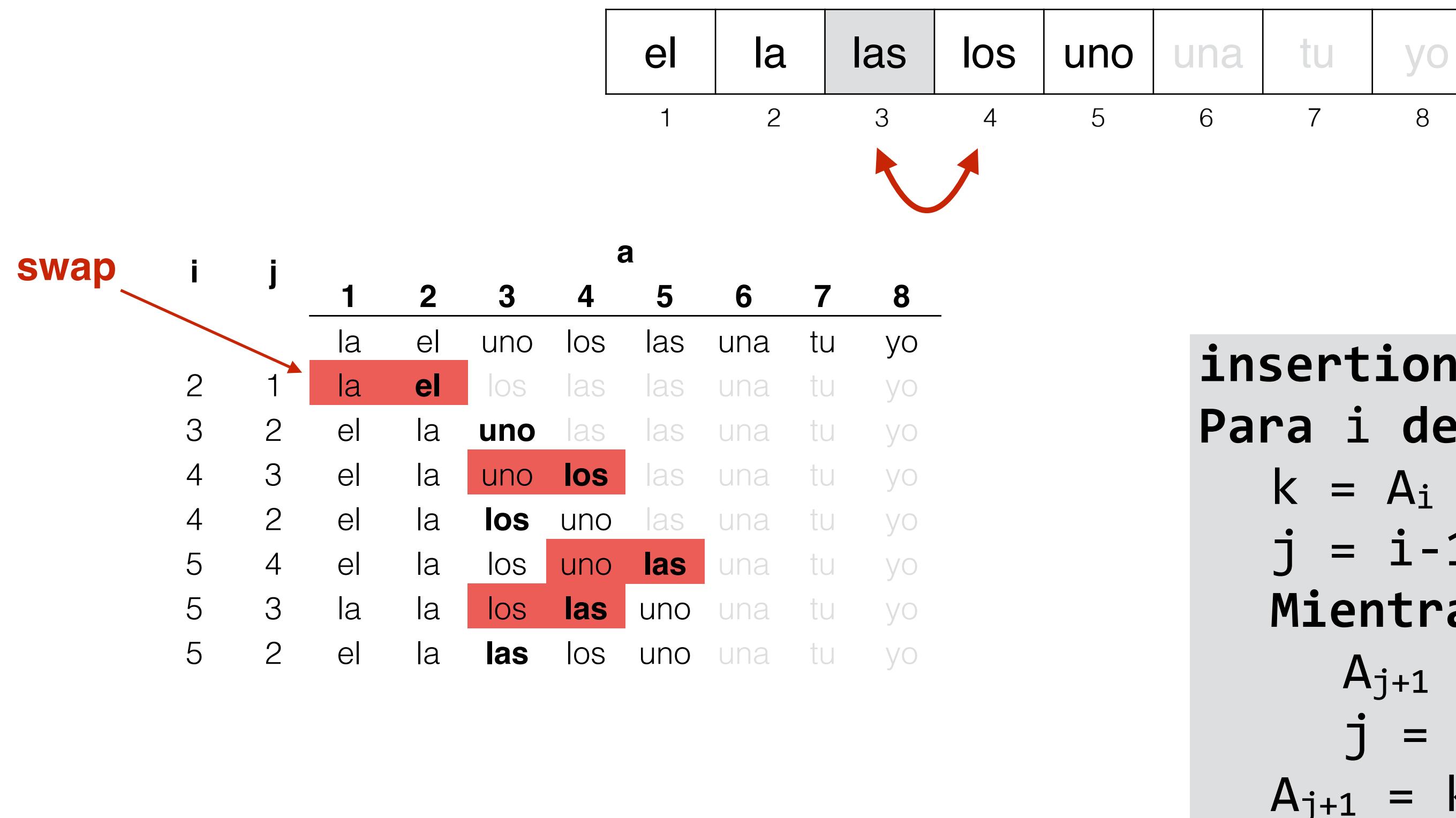
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo

el	la	las	los	uno	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo

el	la	las	los	uno	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo

el	la	las	los	uno	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

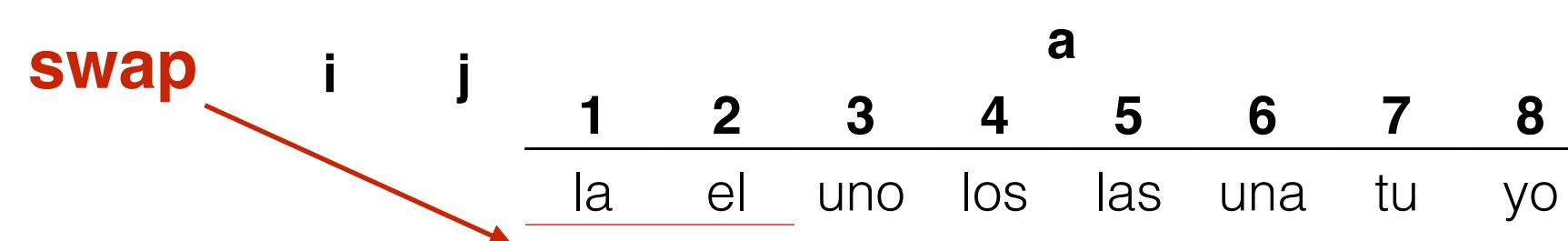
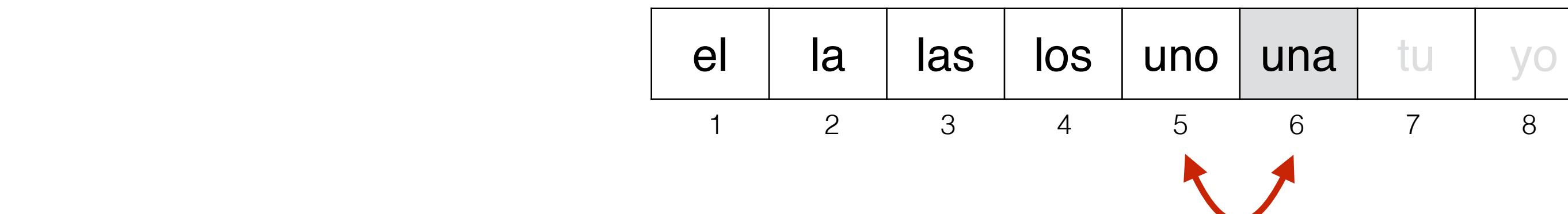
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

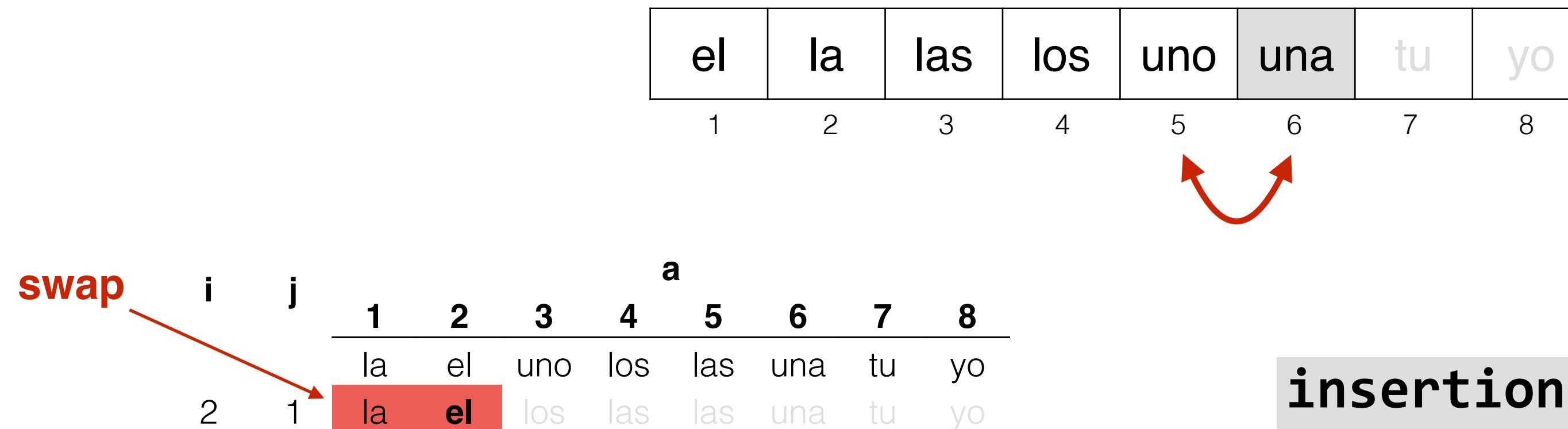
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

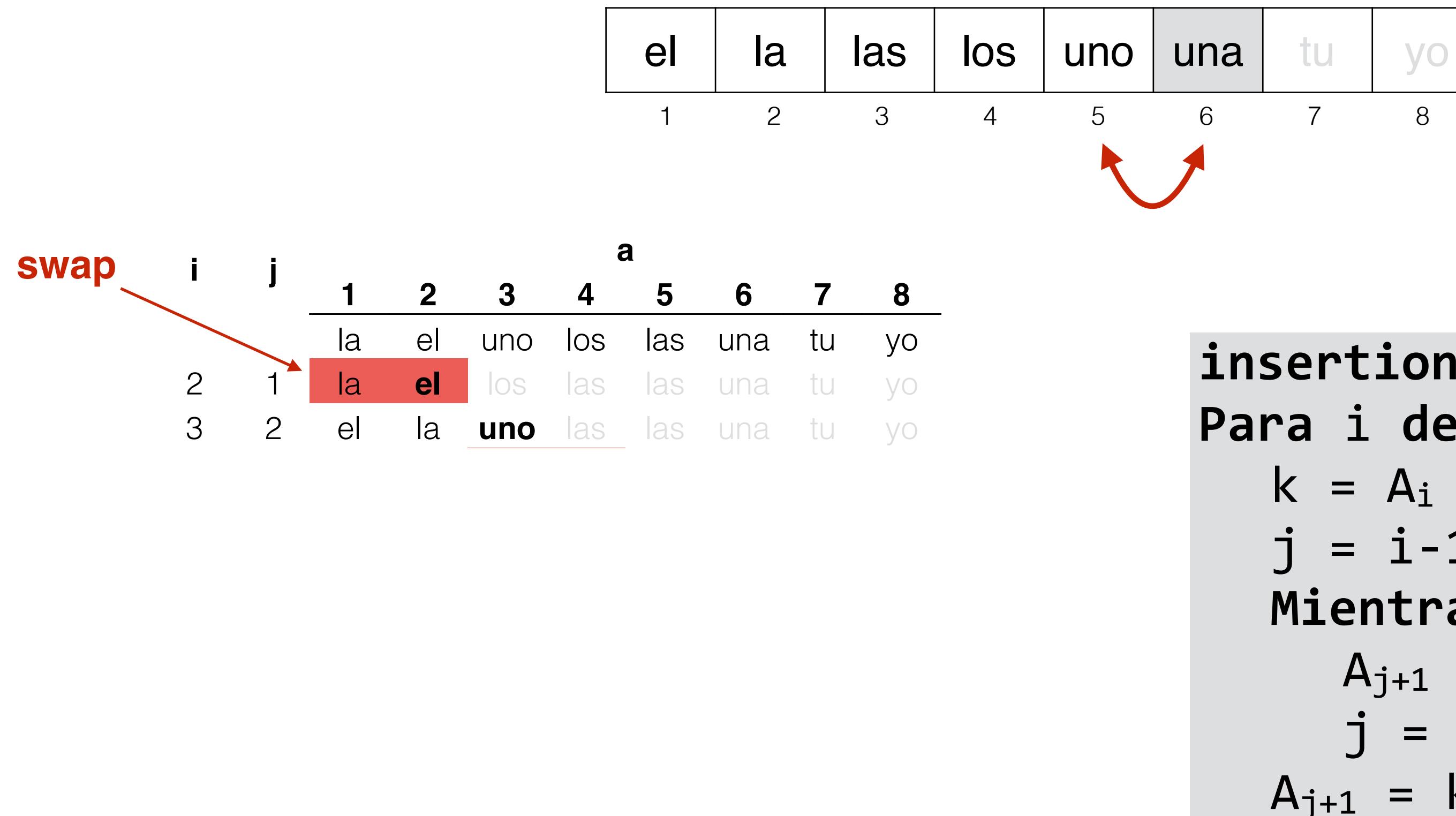
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	las	los	uno	una	tu	yo
	1	2	3	4	5	6	7	8

	i	j	1	2	3	4	5	6	7	8
2	1	1	la	el	uno	los	las	una	tu	yo
3	2	2	el	la	uno	las	las	una	tu	yo
4	3	3	el	la	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

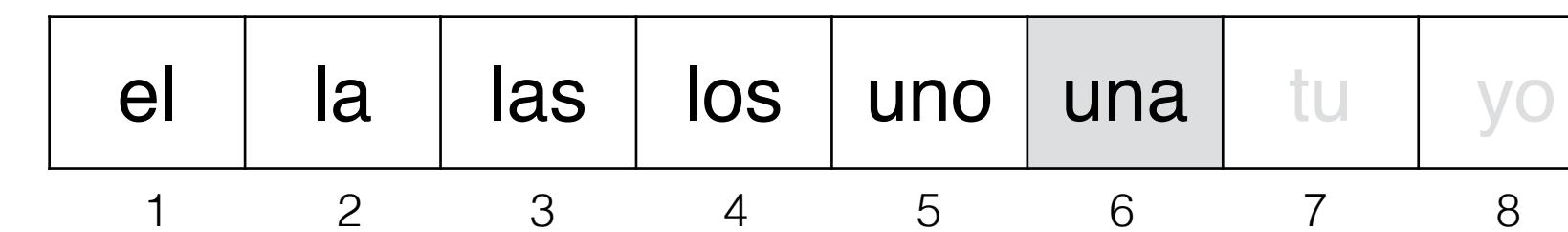
Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo

i j



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

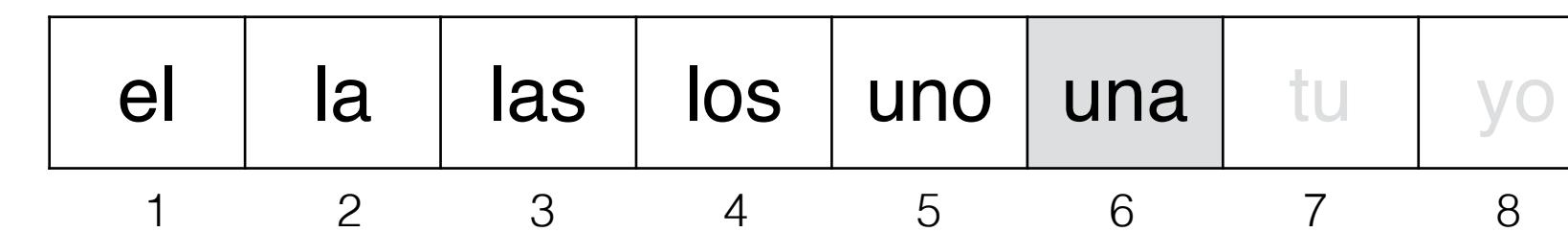
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

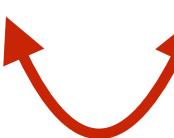
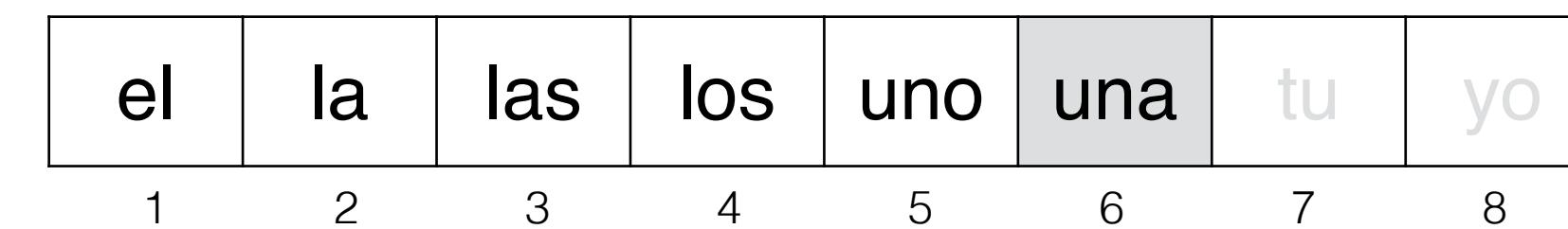
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a							
	1	2	3	4	5	6	7	8
1	el	la	las	los	una	tu	yo	
2	la	el	uno	los	las	una	tu	yo
3	el	la	uno	las	las	una	tu	yo
4	el	la	uno	los	las	una	tu	yo
4	el	la	los	uno	las	una	tu	yo
5	el	la	los	uno	las	una	tu	yo
5	la	la	los	las	uno	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo

el	la	las	los	uno	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo

el	la	las	los	uno	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo

el	la	las	los	uno	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo

el	la	las	los	uno	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo

el	la	las	los	uno	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo

el	la	las	los	uno	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo

el	la	las	los	uno	una	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

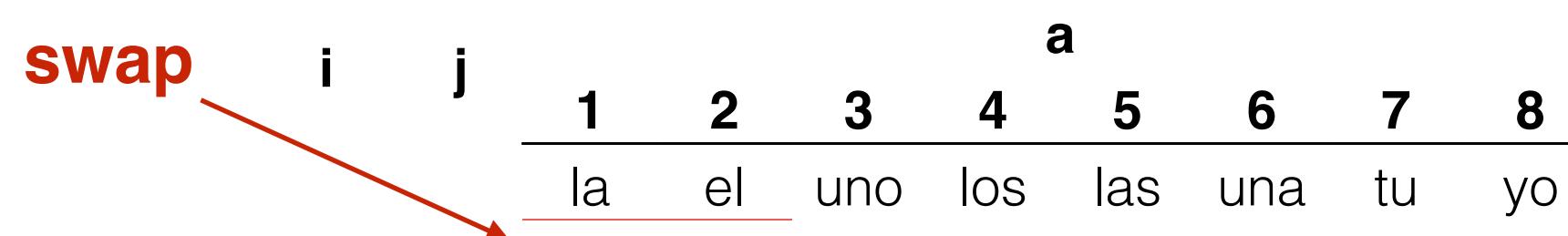
$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

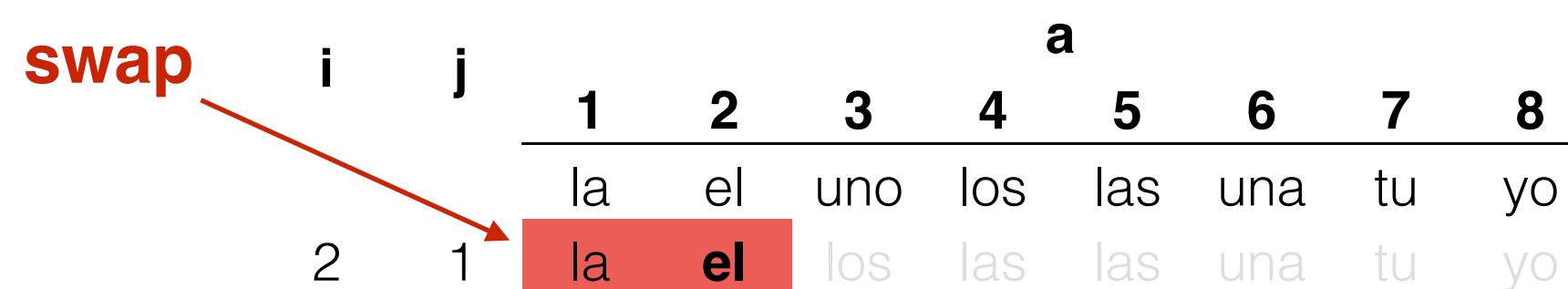
$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8

swap

	i	j	a	1	2	3	4	5	6	7	8
2	1	1	la el uno los las una tu yo	la	el	uno	los	las	una	tu	yo
3	2	2	el la uno las las una tu yo	el	la	uno	las	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8

	i	j	a	1	2	3	4	5	6	7	8
				la	el	uno	los	las	una	tu	yo
2	1	1	swap	la	el	uno	los	las	una	tu	yo
3	2	2		el	la	uno	las	las	una	tu	yo
4	3	3		el	la	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8

swap	i	j	a							
			1	2	3	4	5	6	7	8
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo
	4	2	el	la	los	uno	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8

	i	j	a	1	2	3	4	5	6	7	8
				la	el	uno	los	las	una	tu	yo
2	1	1	swap	la	el	uno	los	las	una	tu	yo
3	2	2		el	la	uno	las	las	una	tu	yo
4	3	3		el	la	uno	los	las	una	tu	yo
4	2	2		el	la	los	uno	las	una	tu	yo
5	4	4		el	la	los	uno	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8

swap

	i	j	a	1	2	3	4	5	6	7	8
2	1	1	la	el	uno	los	las	una	tu	yo	
3	2	1	el	la	uno	las	las	una	tu	yo	
4	3	1	el	la	uno	los	las	una	tu	yo	
4	2	1	el	la	los	uno	las	una	tu	yo	
5	4	1	el	la	los	uno	las	una	tu	yo	
5	3	1	la	la	los	los	las	uno	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8

swap → $i = 2$, $j = 1$

			a	1	2	3	4	5	6	7	8
2	1	la	el	uno	los	las	una	tu	yo		
3	2	el	la	uno	las	las	una	tu	yo		
4	3	el	la	uno	los	las	una	tu	yo		
4	2	el	la	los	uno	las	una	tu	yo		
5	4	el	la	los	uno	los	una	tu	yo		
5	3	la	la	los	los	las	uno	una	tu	yo	
5	2	el	la	las	los	uno	uno	una	tu	yo	

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8

swap → $i = 1$, $j = 2$

a								
	1	2	3	4	5	6	7	8
2	la	el	uno	los	las	una	tu	yo
3	el	la	uno	las	las	una	tu	yo
4	el	la	uno	los	las	una	tu	yo
4	el	la	los	uno	las	una	tu	yo
5	el	la	los	uno	las	una	tu	yo
5	la	la	los	las	uno	una	tu	yo
5	el	la	las	los	uno	una	tu	yo
6	el	la	las	los	uno	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8

swap	i	j	a							
			1	2	3	4	5	6	7	8
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo
	4	2	el	la	los	uno	las	una	tu	yo
	5	4	el	la	los	uno	las	una	tu	yo
	5	3	la	la	los	las	uno	una	tu	yo
	5	2	el	la	las	los	uno	una	tu	yo
	6	5	el	la	las	los	uno	una	tu	yo
	6	4	el	la	las	los	una	uno	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8

swap	i	j	a							
			1	2	3	4	5	6	7	8
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo
	4	2	el	la	los	uno	las	una	tu	yo
	5	4	el	la	los	uno	las	una	tu	yo
	5	3	la	la	los	las	uno	una	tu	yo
	5	2	el	la	las	los	uno	una	tu	yo
	6	5	el	la	las	los	uno	una	tu	yo
	6	4	el	la	las	los	una	uno	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8

swap	i	j	a							
			1	2	3	4	5	6	7	8
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo
	4	2	el	la	los	uno	las	una	tu	yo
	5	4	el	la	los	uno	las	una	tu	yo
	5	3	la	la	los	las	uno	una	tu	yo
	5	2	el	la	las	los	uno	una	tu	yo
	6	5	el	la	las	los	uno	una	tu	yo
	6	4	el	la	las	los	una	uno	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8

swap			a							
	i	j	1	2	3	4	5	6	7	8
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo
	4	2	el	la	los	uno	las	una	tu	yo
	5	4	el	la	los	uno	las	una	tu	yo
	5	3	la	la	los	las	uno	una	tu	yo
	5	2	el	la	las	los	uno	una	tu	yo
	6	5	el	la	las	los	uno	una	tu	yo
	6	4	el	la	las	los	una	uno	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8

swap	i	j	a							
			1	2	3	4	5	6	7	8
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo
	4	2	el	la	los	uno	las	una	tu	yo
	5	4	el	la	los	uno	las	una	tu	yo
	5	3	la	la	los	las	uno	una	tu	yo
	5	2	el	la	las	los	uno	una	tu	yo
	6	5	el	la	las	los	uno	una	tu	yo
	6	4	el	la	las	los	una	uno	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

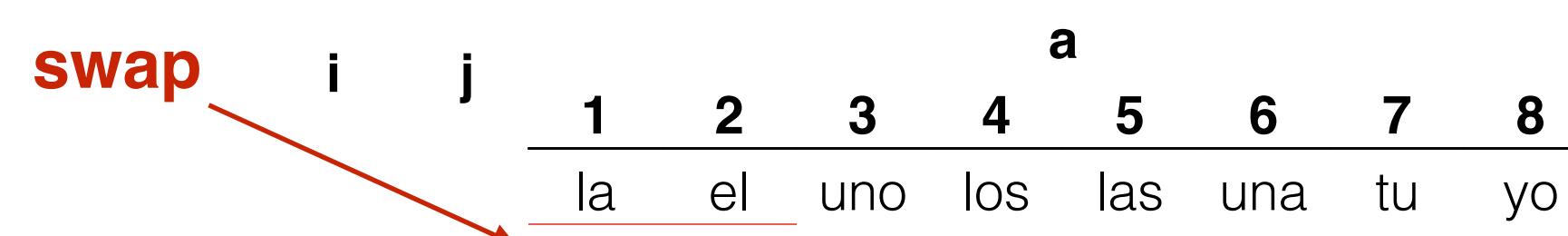
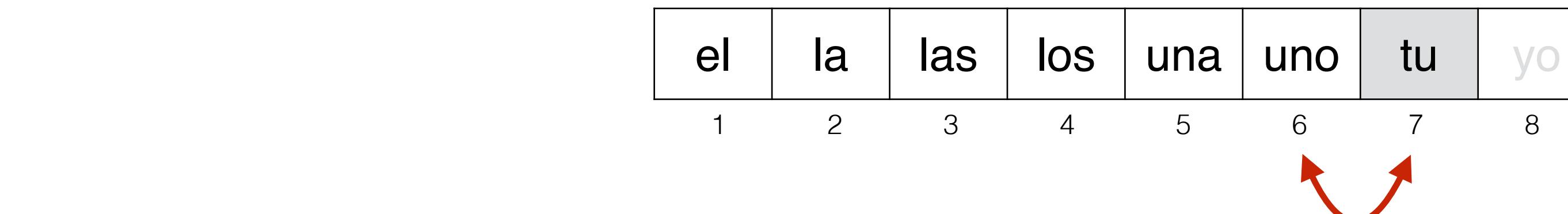
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

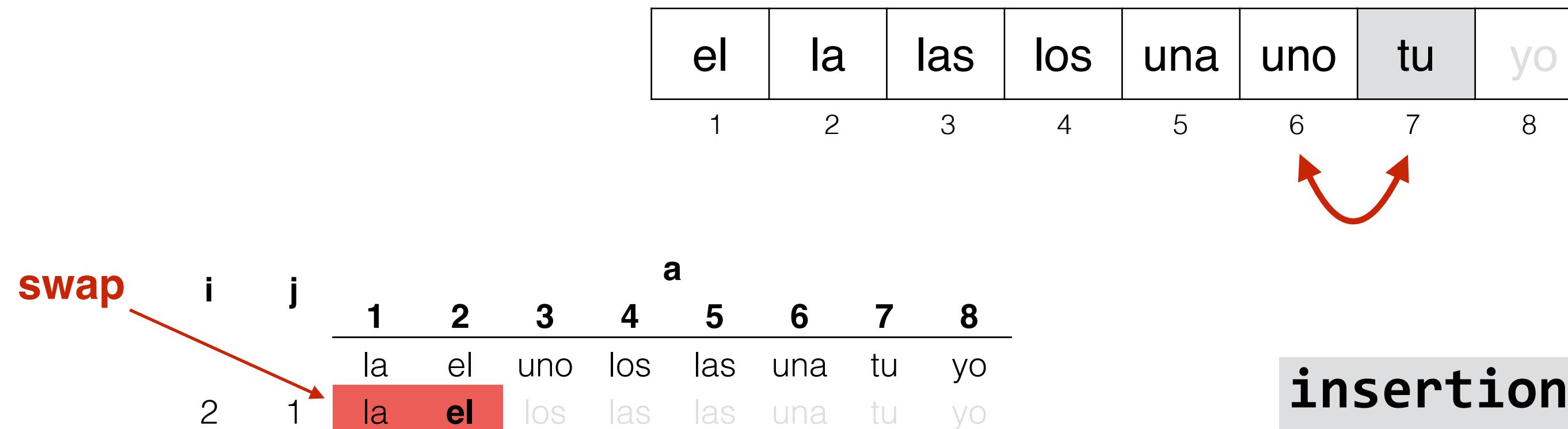
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):
Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

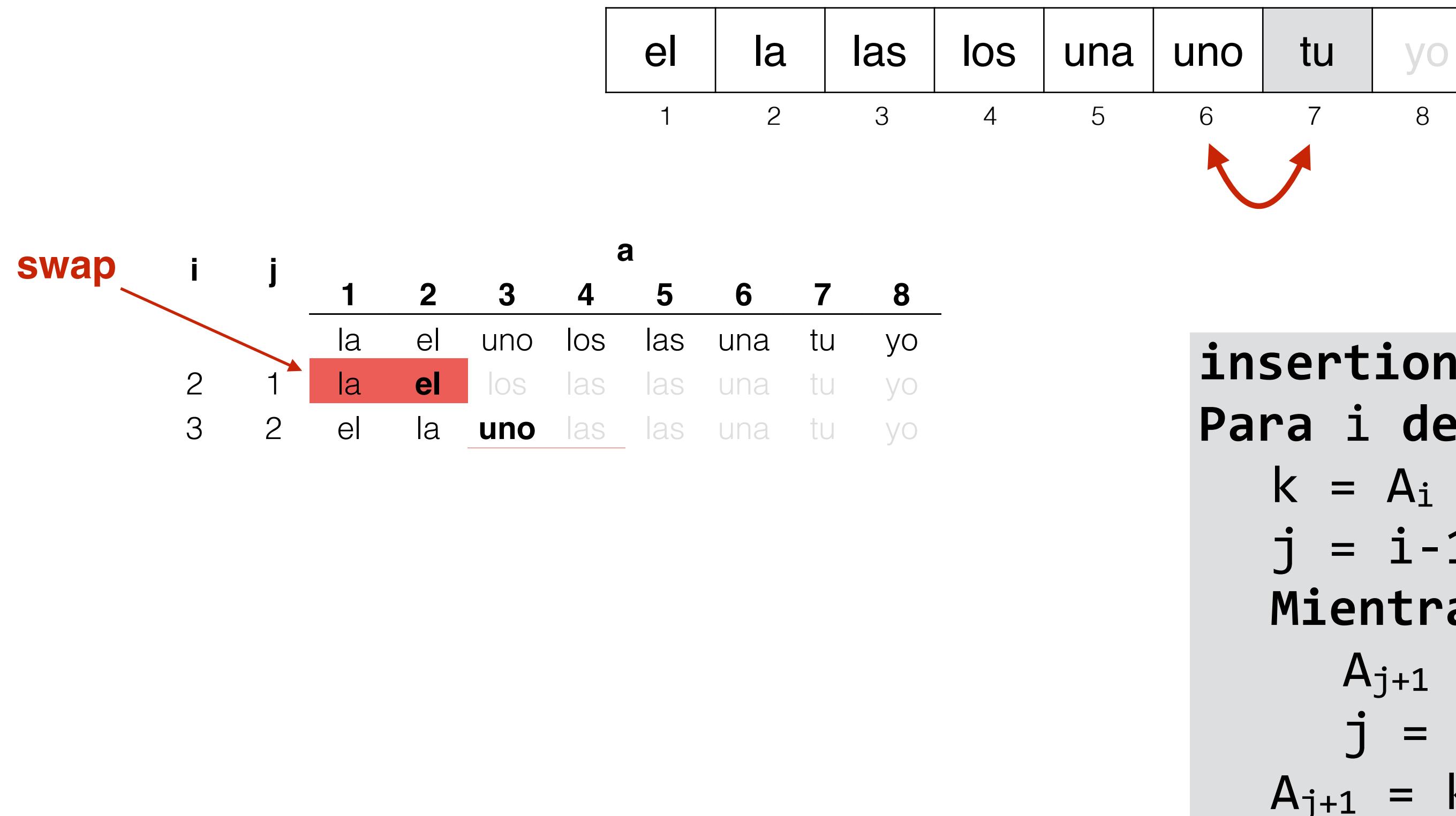
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

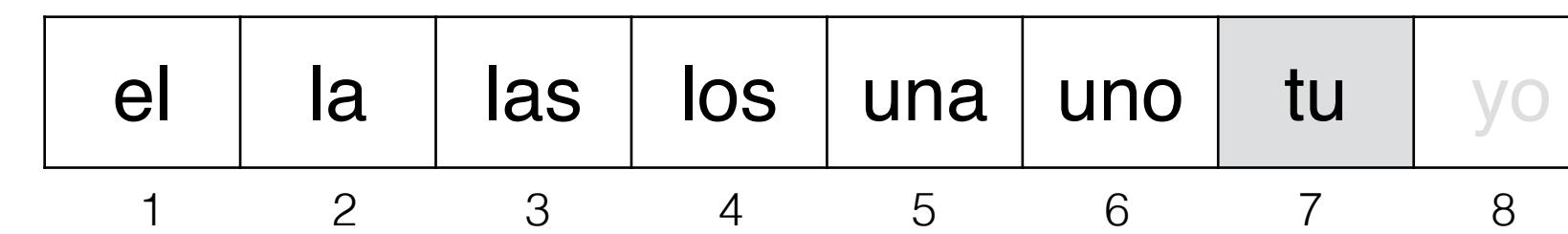
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	1	2	3	4	5	6	7	8	
1	la	el	uno	los	las	una	tu	yo	
2	1	la	el	los	las	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

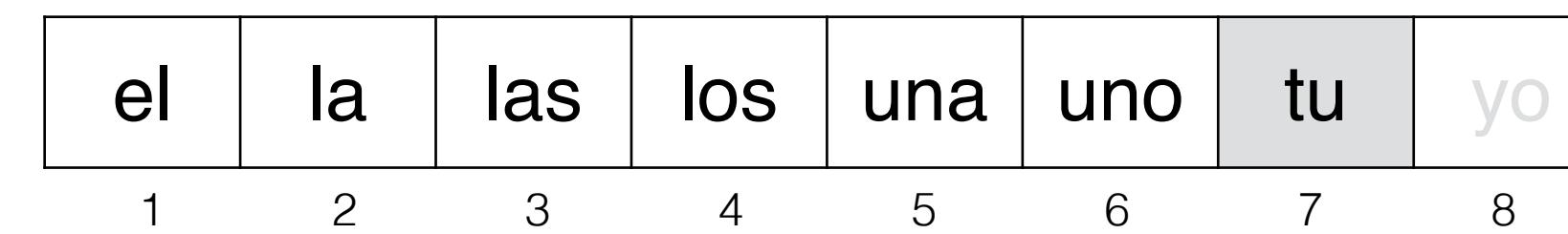
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

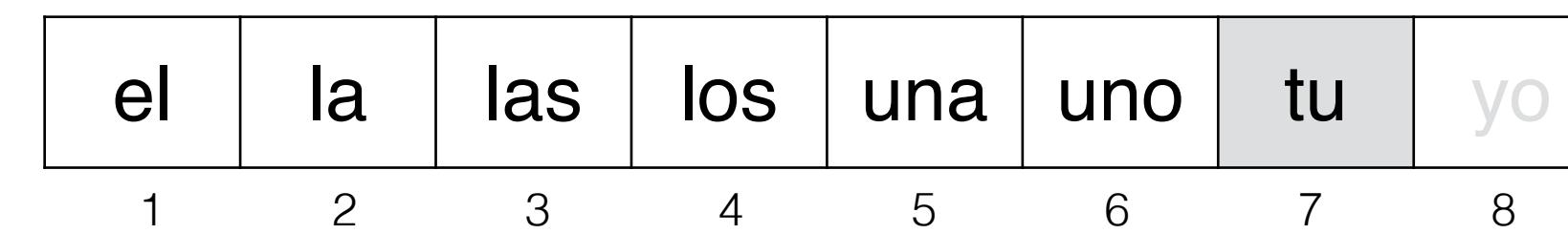
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	las	los	una	uno	tu	yo		
	1	2	3	4	5	6	7	8		
2	1	la	el	uno	los	las	una	tu	yo	
3	2	el	la	uno	las	las	una	tu	yo	
4	3	el	la	uno	los	las	una	tu	yo	
4	2	el	la	los	uno	las	una	tu	yo	
5	4	el	la	los	uno	los	una	tu	yo	
5	3	la	la	los	los	las	uno	una	tu	yo

swap

i

j

a



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

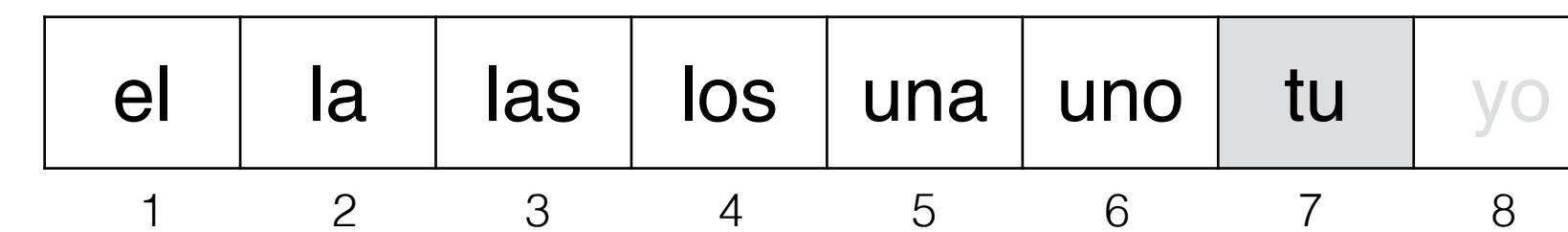
$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	las	los	una	uno	tu	yo		
	1	2	3	4	5	6	7	8		
1	la	el	uno	los	las	una	tu	yo		
2	1	la	el	los	las	una	tu	yo		
3	2	el	la	uno	las	las	una	tu	yo	
4	3	el	la	uno	los	las	una	tu	yo	
4	2	el	la	los	uno	las	una	tu	yo	
5	4	el	la	los	uno	los	una	tu	yo	
5	3	la	la	los	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo	



swap

	i	j	a							
	1	2	3	4	5	6	7	8		
2	1	la	el	uno	los	las	una	tu	yo	
3	2	el	la	uno	las	las	una	tu	yo	
4	3	el	la	uno	los	las	una	tu	yo	
4	2	el	la	los	uno	las	una	tu	yo	
5	4	el	la	los	uno	los	una	tu	yo	
5	3	la	la	los	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo	

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo
6	4	el	la	las	los	una	uno	tu	yo

el	la	las	los	una	uno	tu	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

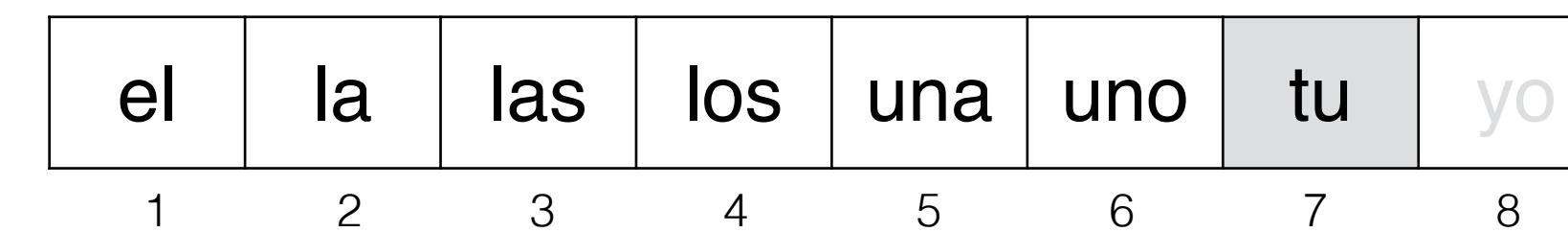
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo
6	4	el	la	las	los	una	uno	tu	yo
7	6	el	la	las	los	una	uno	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

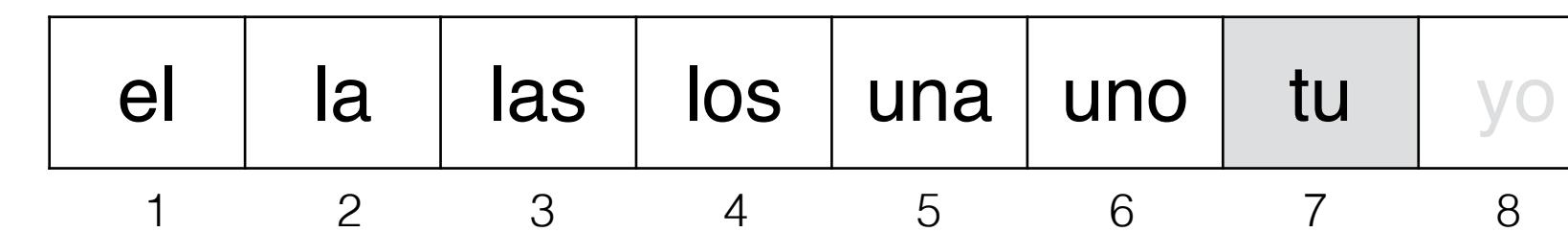
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo
6	4	el	la	las	los	una	uno	tu	yo
7	6	el	la	las	los	una	uno	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

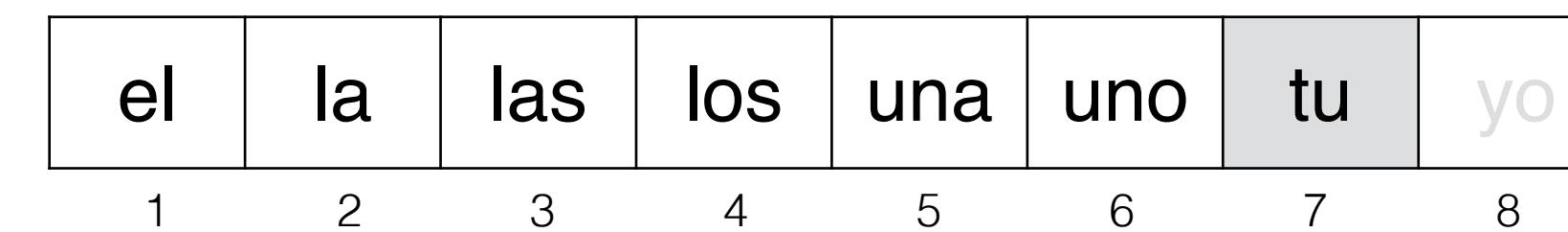
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo
6	4	el	la	las	los	una	uno	tu	yo
7	6	el	la	las	los	una	uno	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

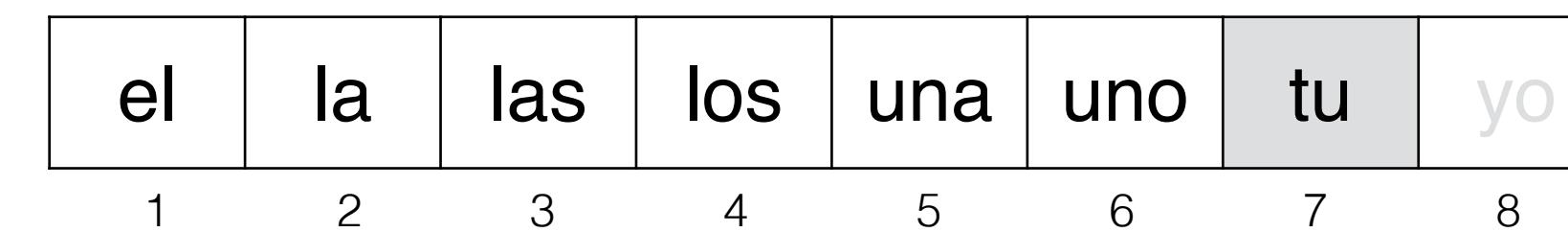
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo
6	4	el	la	las	los	una	uno	tu	yo
7	6	el	la	las	los	una	uno	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

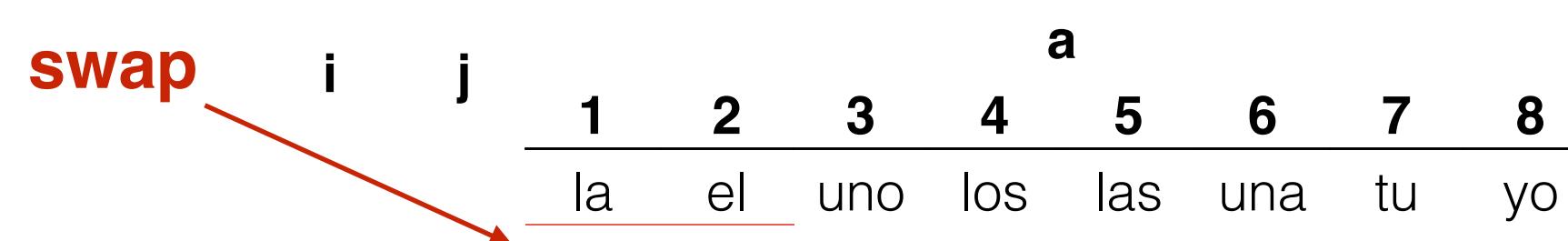
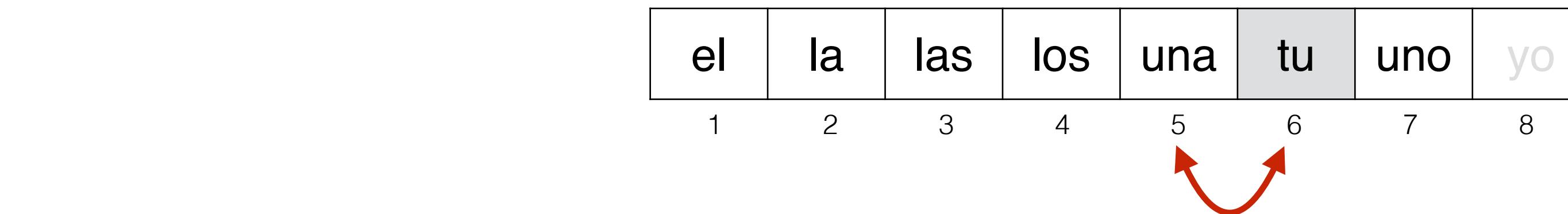
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

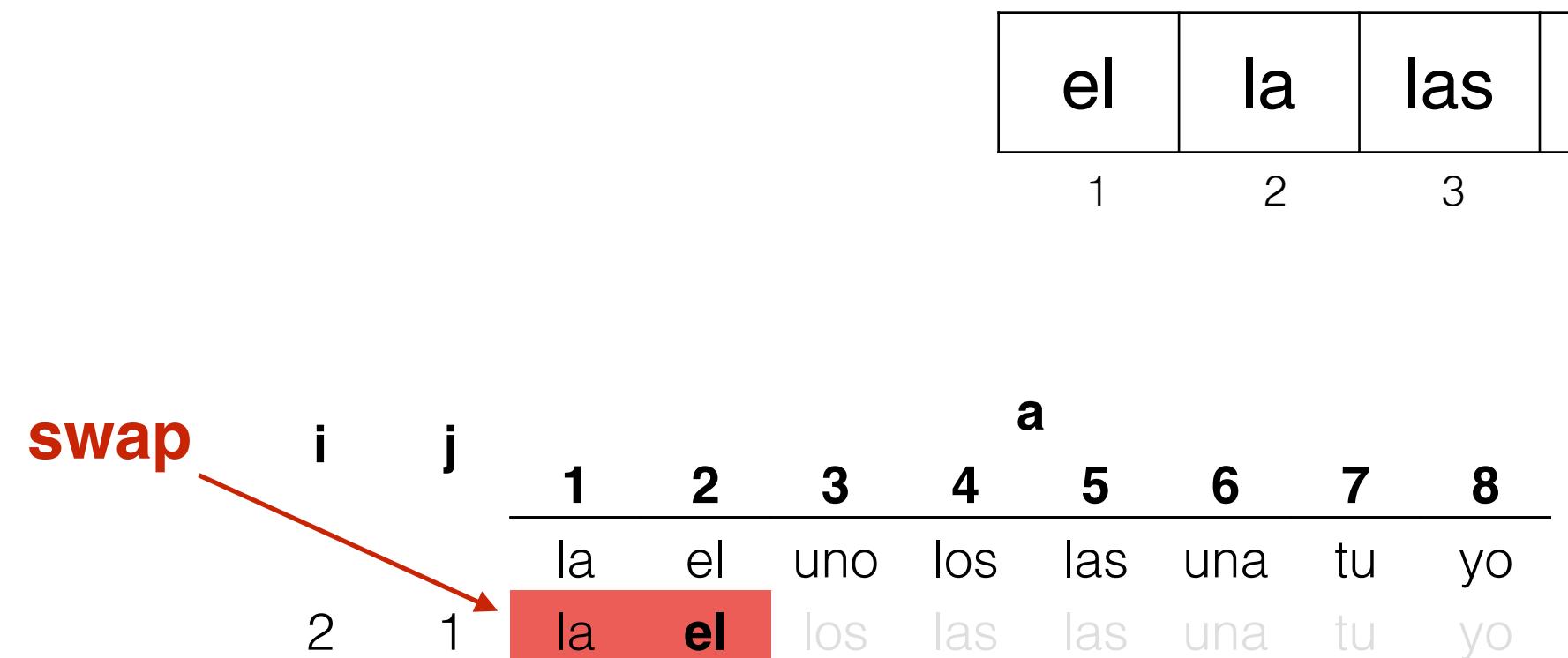
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

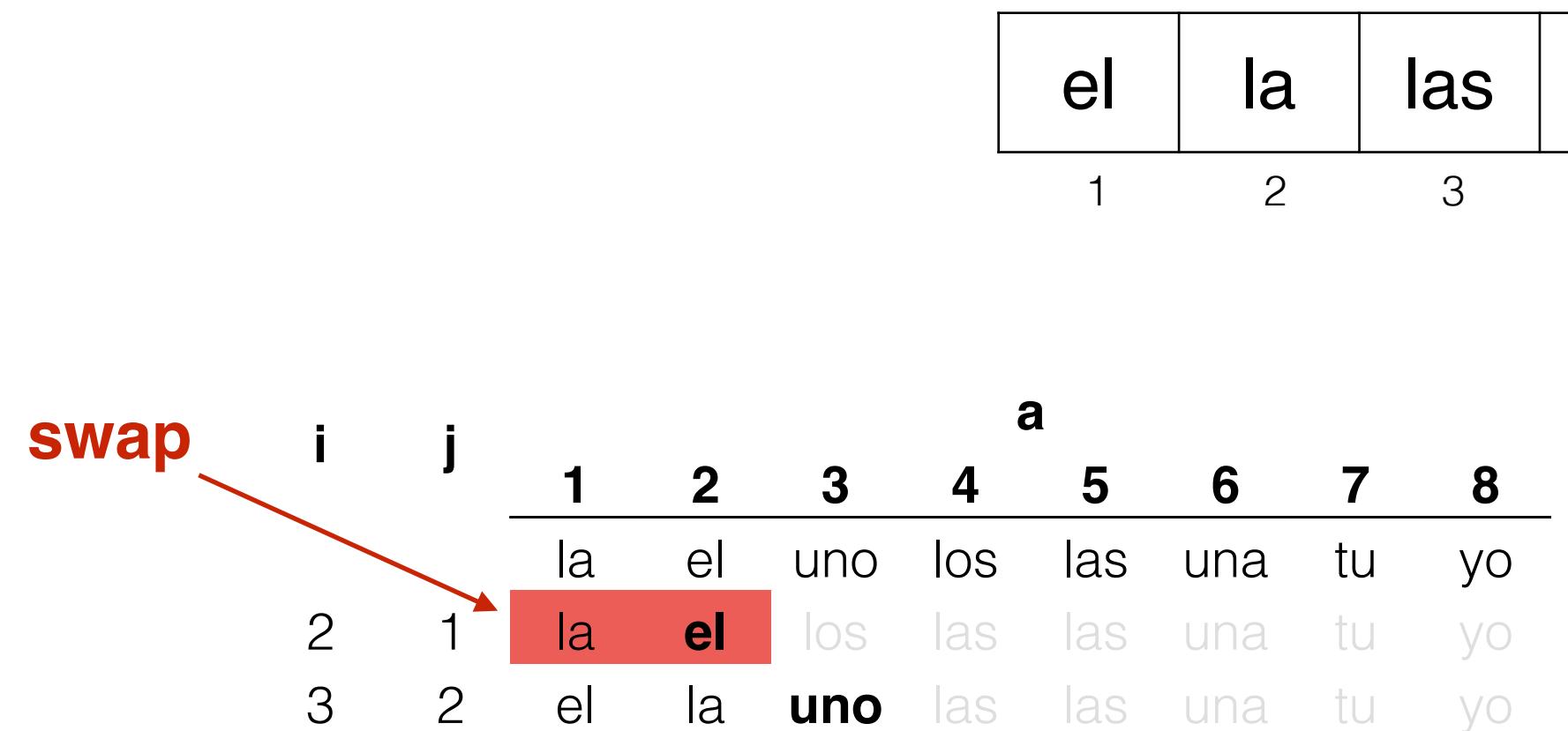
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):
Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

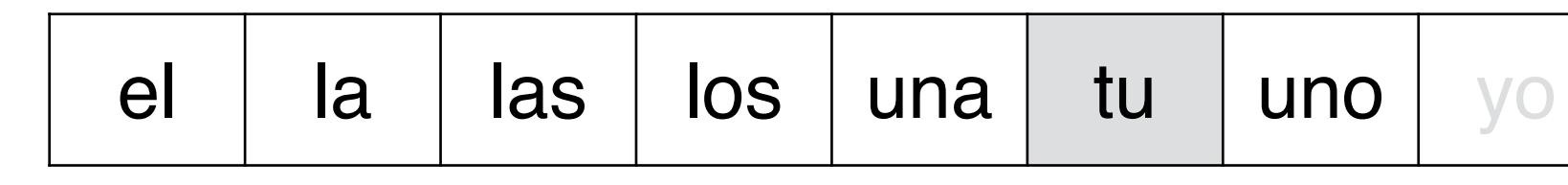
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	1	2	3	4	5	6	7	8	a
i	2	1							
j			2	3	4	5	6	7	
la	el	uno	los	las	una	tu	yo		
2	1	la	el	los	las	una	tu	yo	
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	las	los	una	tu	uno	yo	
	1	2	3	4	5	6	7	8	
swap	i	j							
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

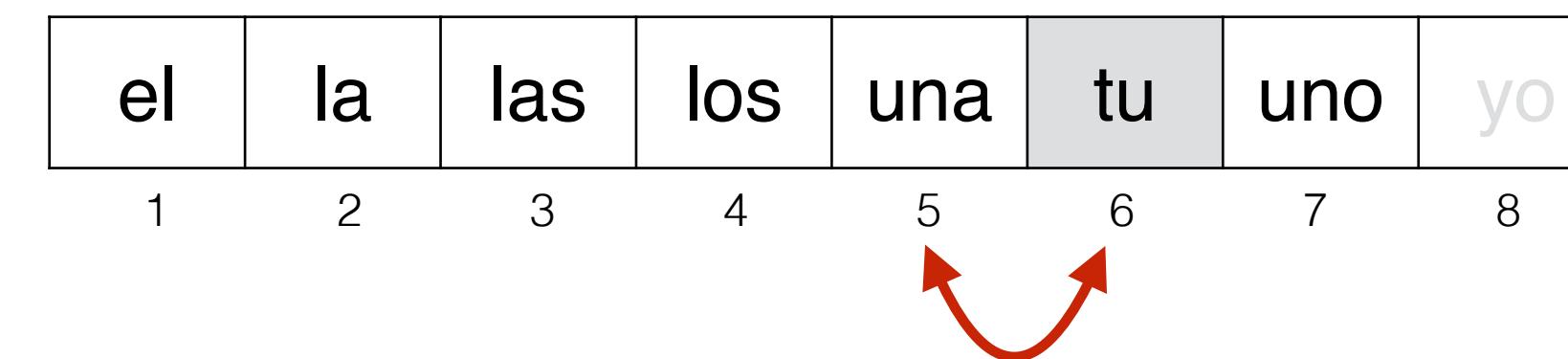
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

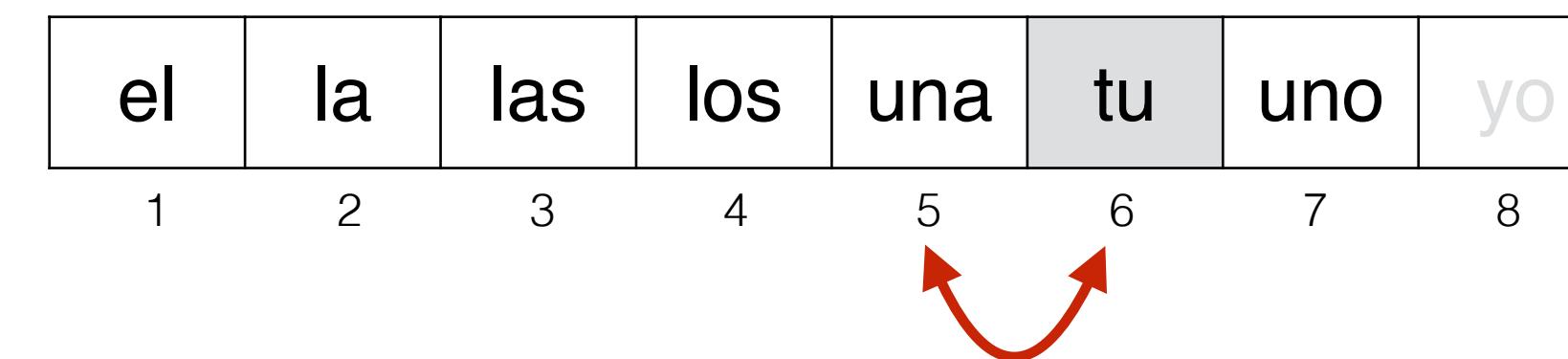
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

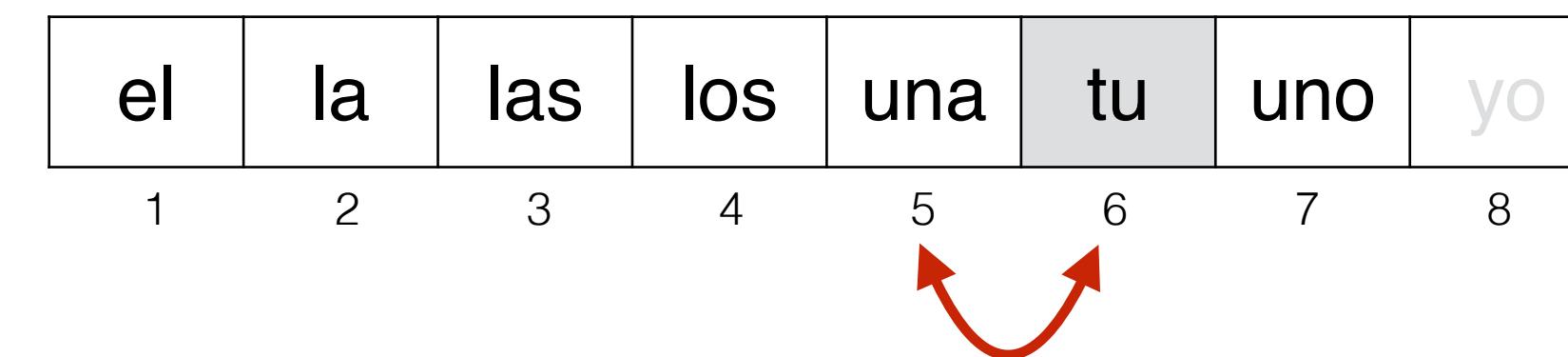
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo

el	la	las	los	una	tu	uno	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap → i j

	a							
	1	2	3	4	5	6	7	8
2	la	el	uno	los	las	una	tu	yo
3	el	la	uno	las	las	una	tu	yo
4	el	la	uno	los	las	una	tu	yo
4	el	la	los	uno	las	una	tu	yo
5	el	la	los	uno	las	una	tu	yo
5	la	la	los	las	uno	una	tu	yo
5	el	la	los	las	uno	una	tu	yo
6	el	la	las	los	uno	una	tu	yo
6	el	la	las	los	una	uno	tu	yo
6	el	la	las	los	una	uno	tu	yo

el	la	las	los	una	tu	uno	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap → i j

	a							
	1	2	3	4	5	6	7	8
2	la	el	uno	los	las	una	tu	yo
3	el	la	uno	las	las	una	tu	yo
4	el	la	uno	los	las	una	tu	yo
4	el	la	los	uno	las	una	tu	yo
5	el	la	los	uno	las	una	tu	yo
5	la	la	los	las	uno	una	tu	yo
5	el	la	las	uno	una	una	tu	yo
6	el	la	las	los	uno	una	tu	yo
6	el	la	las	los	una	uno	tu	yo
7	el	la	las	los	una	uno	tu	yo

el	la	las	los	una	tu	uno	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

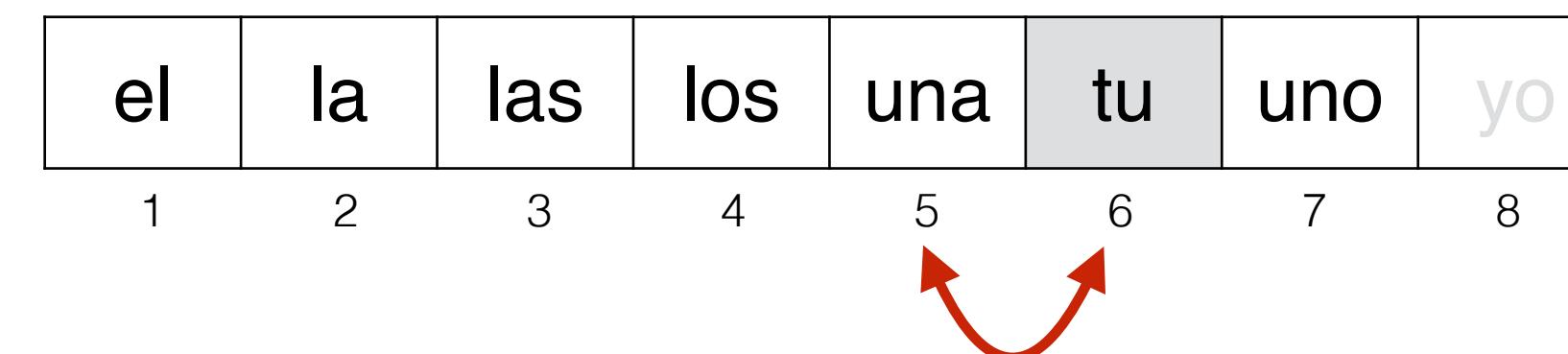
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo
6	4	el	la	las	los	una	uno	tu	yo
7	6	el	la	las	los	una	uno	tu	yo
7	5	el	la	las	los	una	tu	uno	yo



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

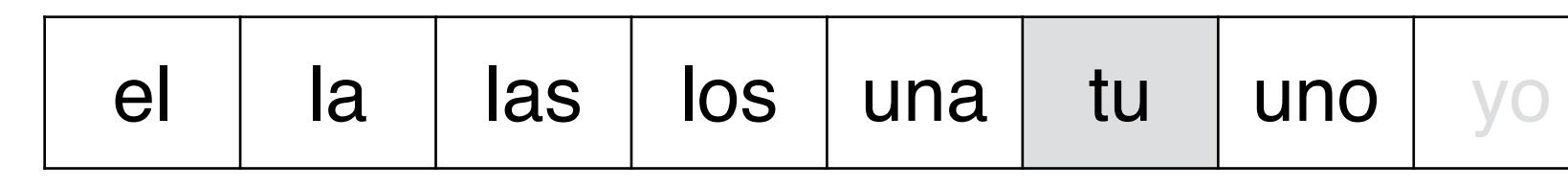
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo
6	4	el	la	las	los	una	uno	tu	yo
7	6	el	la	las	los	una	uno	tu	yo
7	5	el	la	las	los	una	tu	uno	yo



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

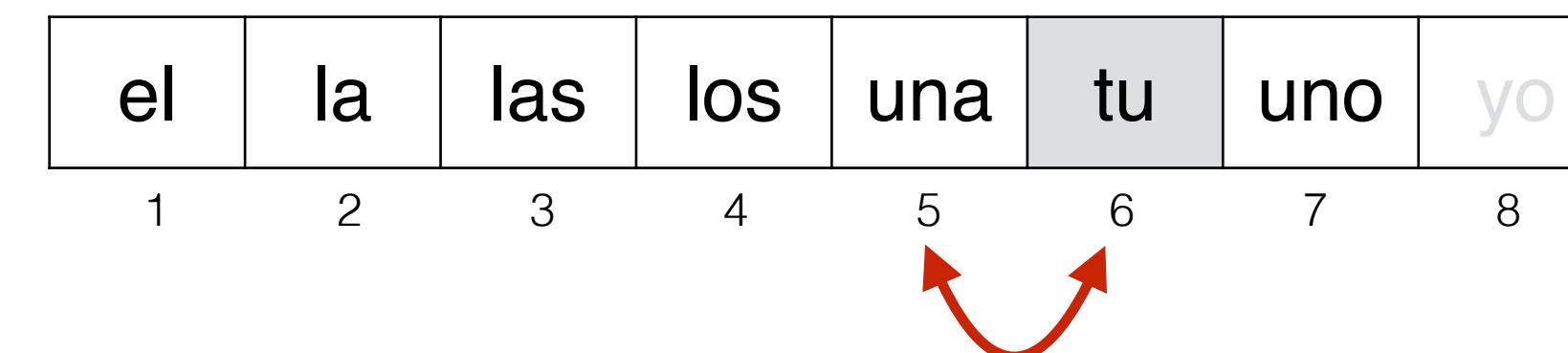
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo
6	4	el	la	las	los	una	uno	tu	yo
7	6	el	la	las	los	una	uno	tu	yo
7	5	el	la	las	los	una	tu	uno	yo



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

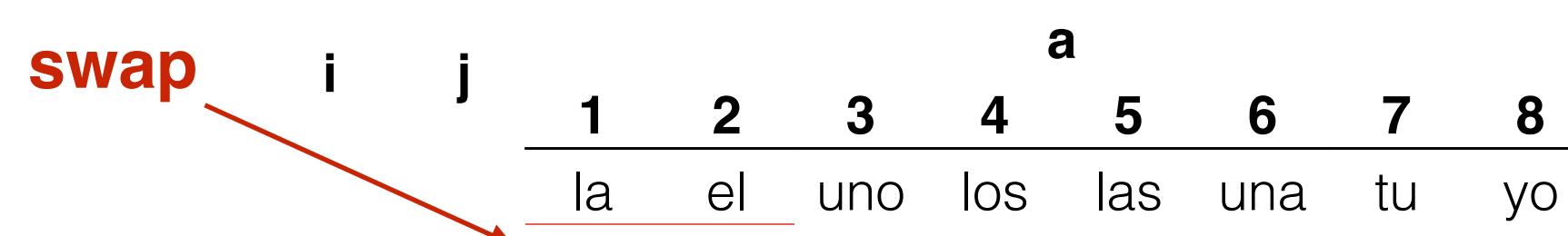
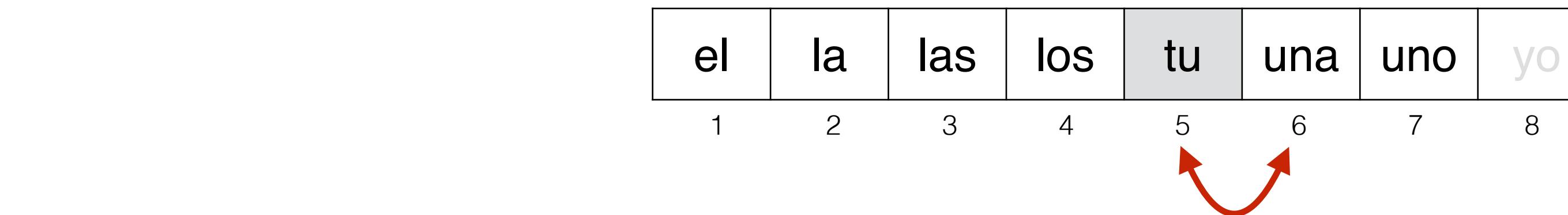
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

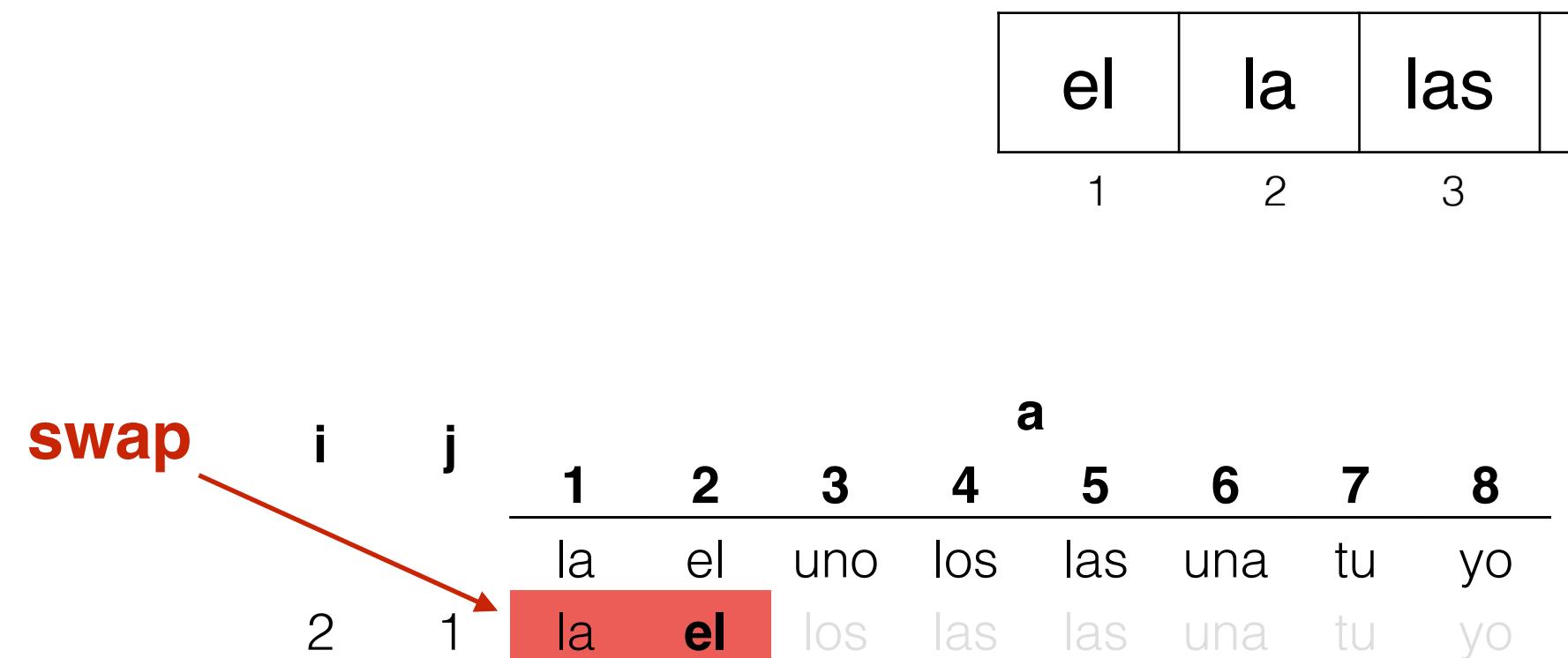
$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.



insertionsort(A):
Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

$i \quad j$

a

	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	<u>uno</u>	<u>los</u>	<u>las</u>	<u>una</u>	<u>tu</u>	<u>yo</u>

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	las	los	tu	una	uno	yo	
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

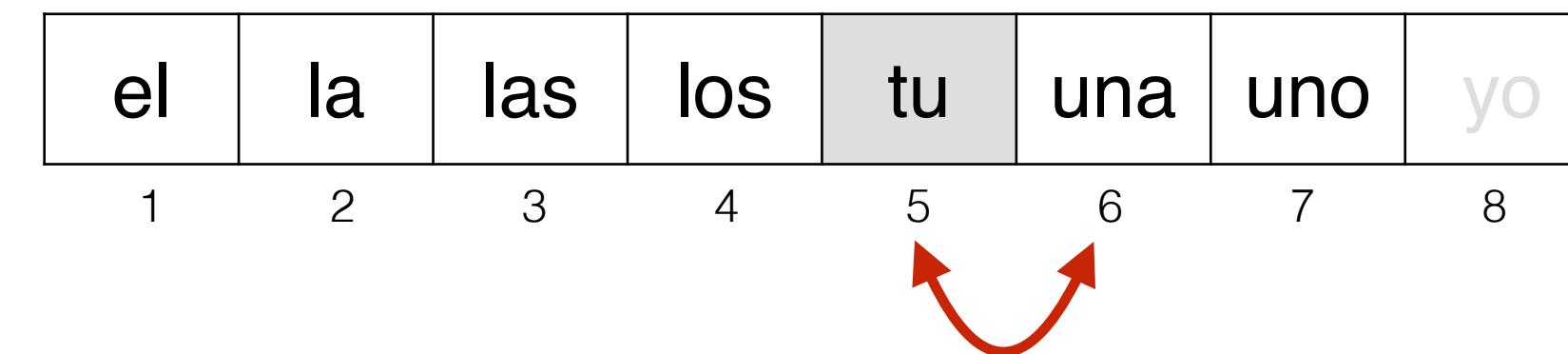
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a							
	1	2	3	4	5	6	7	8
2	la	el	uno	los	las	una	tu	yo
3	el	la	uno	las	las	una	tu	yo
4	el	la	uno	los	las	una	tu	yo
4	el	la	los	uno	las	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

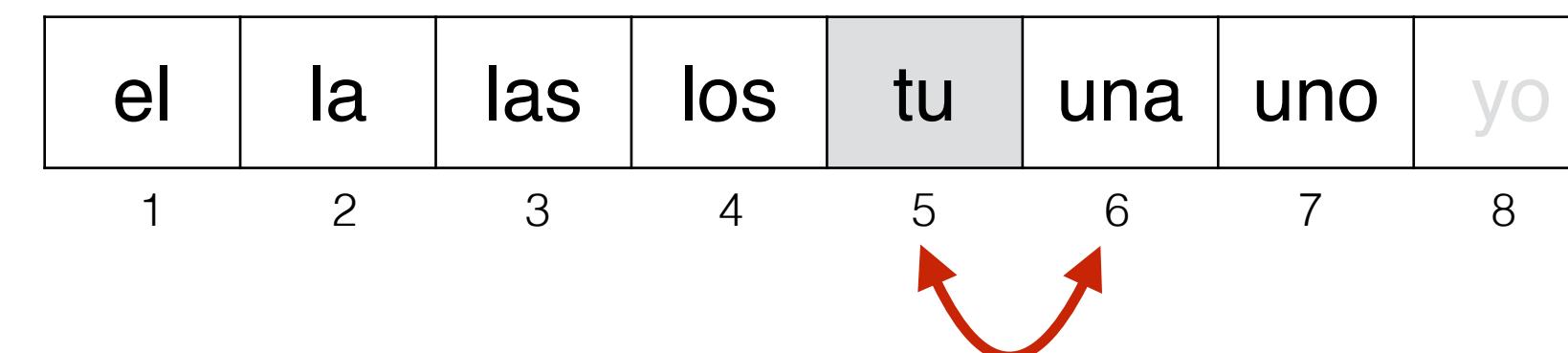
Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo

i j



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

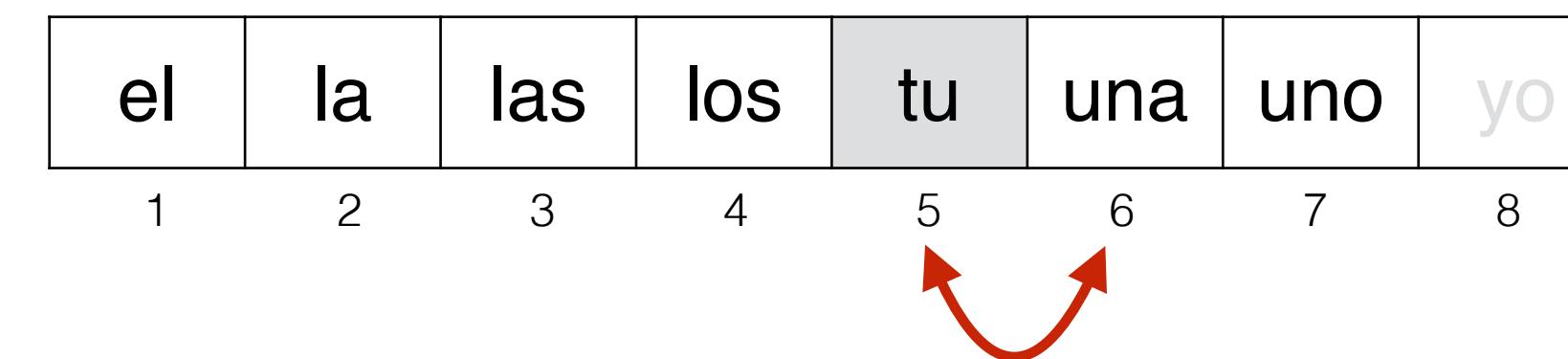
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

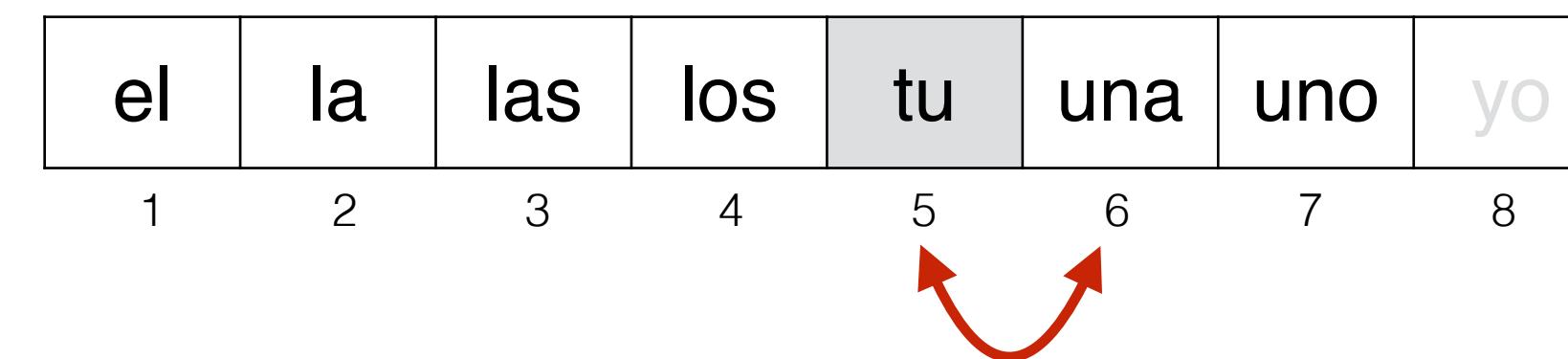
$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	las	los	tu	una	uno	yo
	1	2	3	4	5	6	7	8
1	la	el	uno	los	las	una	tu	yo
2	1	la	el	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu
4	3	el	la	uno	los	las	una	tu
4	2	el	la	los	uno	las	una	tu
5	4	el	la	los	uno	las	una	tu
5	3	la	la	los	las	uno	una	tu
5	2	el	la	las	los	uno	una	tu



	i	j	a	1	2	3	4	5	6	7	8
	1	2	3	4	5	6	7	8			
1	la	el	uno	los	las	una	tu	yo			
2	1	la	el	los	las	una	tu	yo			
3	2	el	la	uno	las	las	una	tu	yo		
4	3	el	la	uno	los	las	una	tu	yo		
4	2	el	la	los	uno	las	una	tu	yo		
5	4	el	la	los	uno	las	una	tu	yo		
5	3	la	la	los	las	uno	una	tu	yo		
5	2	el	la	las	los	uno	una	tu	yo		

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

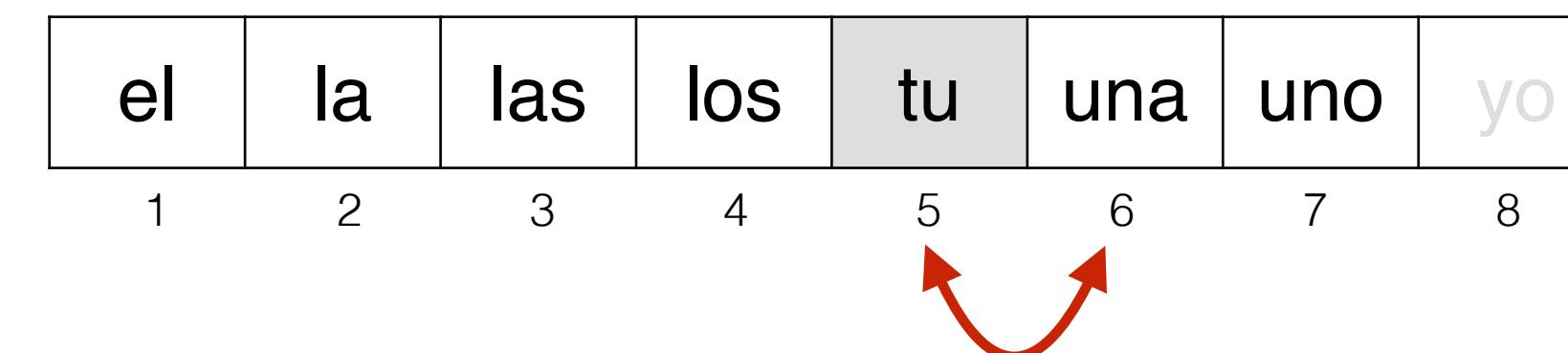
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

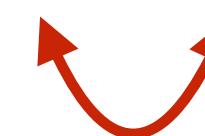
Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	a								
	1	2	3	4	5	6	7	8	
2	1	la	el	uno	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo
4	3	el	la	uno	los	las	una	tu	yo
4	2	el	la	los	uno	las	una	tu	yo
5	4	el	la	los	uno	las	una	tu	yo
5	3	la	la	los	las	uno	una	tu	yo
5	2	el	la	las	los	uno	una	tu	yo
6	5	el	la	las	los	uno	una	tu	yo
6	4	el	la	las	los	una	uno	tu	yo

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap → i j

	a							
	1	2	3	4	5	6	7	8
2	la	el	uno	los	las	una	tu	yo
3	el	la	uno	las	las	una	tu	yo
4	el	la	uno	los	las	una	tu	yo
4	el	la	los	uno	las	una	tu	yo
5	el	la	los	uno	las	una	tu	yo
5	la	la	los	las	uno	una	tu	yo
5	el	la	las	los	uno	una	tu	yo
6	el	la	las	los	uno	una	tu	yo
6	el	la	las	los	una	uno	tu	yo
7	el	la	las	los	una	uno	tu	yo

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

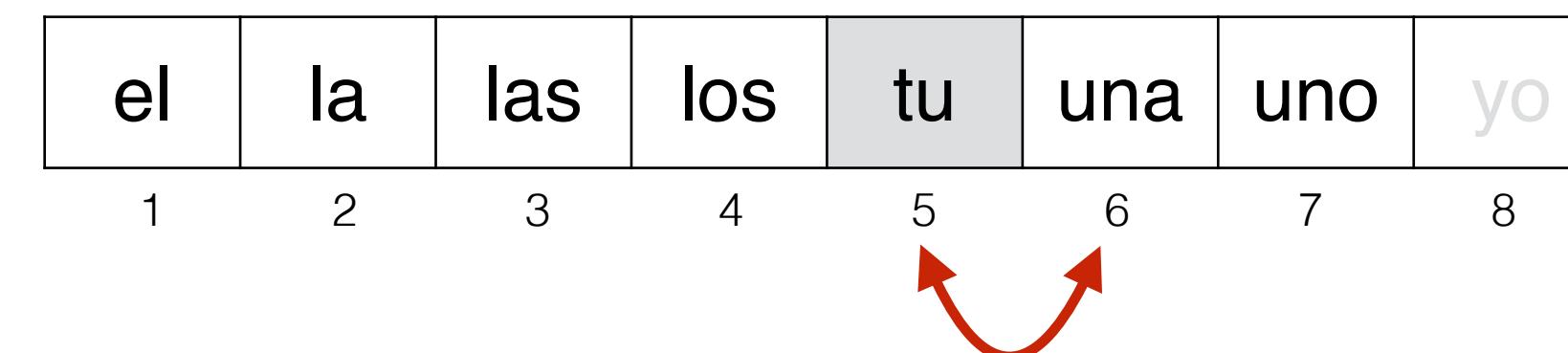
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

swap

	i	j	a							
	1	2	3	4	5	6	7	8		
2	1	la	el	uno	los	las	una	tu	yo	
3	2	el	la	uno	las	las	una	tu	yo	
4	3	el	la	uno	los	las	una	tu	yo	
4	2	el	la	los	uno	las	una	tu	yo	
5	4	el	la	los	uno	las	una	tu	yo	
5	3	la	la	los	las	uno	una	tu	yo	
5	2	el	la	las	los	uno	una	tu	yo	
6	5	el	la	las	los	uno	una	tu	yo	
6	4	el	la	las	los	una	uno	tu	yo	
7	6	el	la	las	los	una	uno	tu	yo	
7	5	el	la	las	los	una	tu	uno	yo	



insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

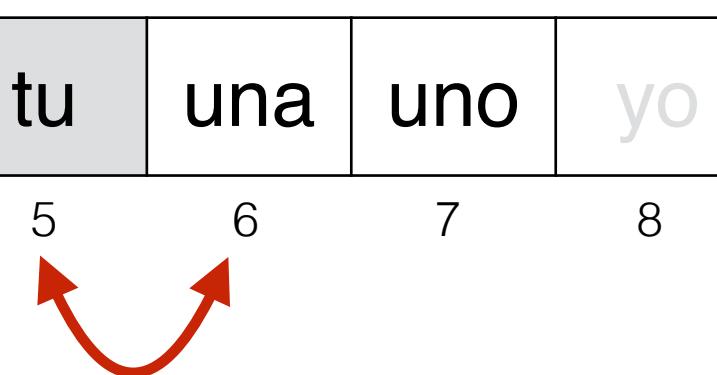
$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	las	los	tu	una	uno	yo
	1	2	3	4	5	6	7	8
1	la	el	uno	los	las	una	tu	yo
2	1	la	el	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu
4	3	el	la	uno	los	las	una	tu
4	2	el	la	los	uno	las	una	tu
5	4	el	la	los	uno	las	una	tu
5	3	la	la	los	las	uno	una	tu
5	2	el	la	las	los	uno	una	tu
6	5	el	la	las	los	uno	una	tu
6	4	el	la	las	los	una	uno	tu
7	6	el	la	las	los	una	uno	tu
7	5	el	la	las	los	una	tu	uno
7	4	el	la	las	los	tu	una	uno



	i	j	a	1	2	3	4	5	6	7	8
	1	2	3	4	5	6	7	8			
1	la	el	uno	los	las	una	tu	yo			
2	1	la	el	los	las	una	tu	yo			
3	2	el	la	uno	las	las	una	tu			
4	3	el	la	uno	los	las	una	tu			
4	2	el	la	los	uno	las	una	tu			
5	4	el	la	los	uno	las	una	tu			
5	3	la	la	los	las	uno	una	tu			
5	2	el	la	las	los	uno	una	tu			
6	5	el	la	las	los	uno	una	tu			
6	4	el	la	las	los	una	uno	tu			
7	6	el	la	las	los	una	uno	tu			
7	5	el	la	las	los	una	tu	uno			
7	4	el	la	las	los	tu	una	uno			

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

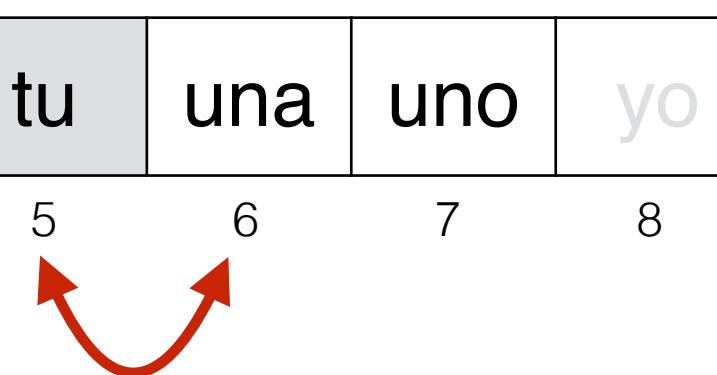
$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

	el	la	las	los	tu	una	uno	yo
	1	2	3	4	5	6	7	8
1	la	el	uno	los	las	una	tu	yo
2	1	la	el	los	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu
4	3	el	la	uno	los	las	una	tu
4	2	el	la	los	uno	las	una	tu
5	4	el	la	los	uno	las	una	tu
5	3	la	la	los	las	uno	una	tu
5	2	el	la	las	los	uno	una	tu
6	5	el	la	las	los	uno	una	tu
6	4	el	la	las	los	una	uno	tu
7	6	el	la	las	los	una	uno	tu
7	5	el	la	las	los	una	tu	uno
7	4	el	la	las	los	tu	una	uno



	i	j	a	1	2	3	4	5	6	7	8
	1	2	3	4	5	6	7	8			
1	la	el	uno	los	las	una	tu	yo			
2	1	la	el	los	las	una	tu	yo			
3	2	el	la	uno	las	las	una	tu			
4	3	el	la	uno	los	las	una	tu			
4	2	el	la	los	uno	las	una	tu			
5	4	el	la	los	uno	las	una	tu			
5	3	la	la	los	las	uno	una	tu			
5	2	el	la	las	los	uno	una	tu			
6	5	el	la	las	los	uno	una	tu			
6	4	el	la	las	los	una	uno	tu			
7	6	el	la	las	los	una	uno	tu			
7	5	el	la	las	los	una	tu	uno			
7	4	el	la	las	los	tu	una	uno			

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

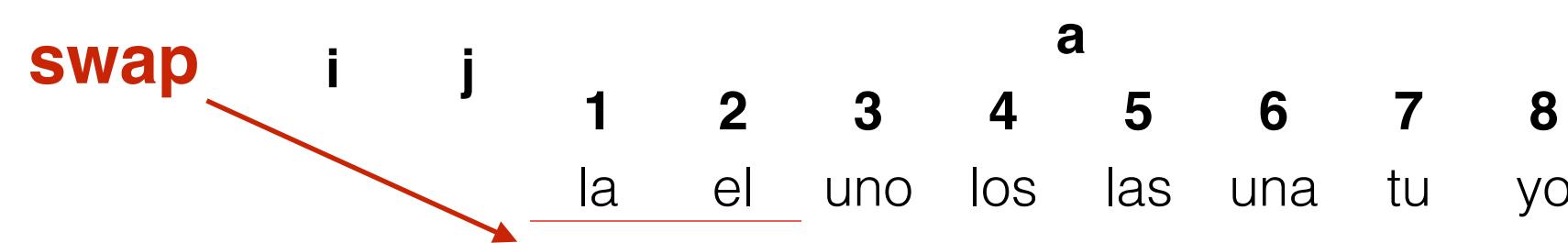
$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8



insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8



insertionsort(A):
Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8

		a								
		1	2	3	4	5	6	7	8	
swap	i	la	el	uno	los	las	una	tu	yo	
	j	la	el	uno	los	las	una	tu	yo	
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	los	las	una	tu	yo

insertionsort(A):
Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8

	i	j	1	2	3	4	5	6	7	8	a
2	1	1	la	el	uno	los	las	una	tu	yo	
3	2	2	el	la	uno	las	las	una	tu	yo	
4	3	3	el	la	uno	los	las	una	tu	yo	

swap

i

j

insertionsort(A):

Para i de 2 a N:

$k = A_i$

$j = i - 1$

Mientras $j \geq 1$ and $A_j > k$:

$A_{j+1} = A_j$

$j = j - 1$

$A_{j+1} = k$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8

	i	j	1	2	3	4	5	6	7	8
			la	el	uno	los	las	una	tu	yo
2	1	la	el	uno	los	las	las	una	tu	yo
3	2	el	la	uno	las	las	una	tu	yo	
4	3	el	la	uno	los	las	una	tu	yo	
4	2	el	la	los	uno	las	una	tu	yo	

swap

i
2
1

j
1
2
3
4
2
3
4
2

insertionsort(A):

Para i de 2 a N:

$k = A_i$

$j = i - 1$

Mientras $j \geq 1$ and $A_j > k$:

$A_{j+1} = A_j$

$j = j - 1$

$A_{j+1} = k$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8

	i	j	a	1	2	3	4	5	6	7	8
				la	el	uno	los	las	una	tu	yo
2	1	1	swap	la	el	uno	los	las	una	tu	yo
3	2	2		el	la	uno	las	las	una	tu	yo
4	3	3		el	la	uno	los	las	una	tu	yo
4	2	2		el	la	los	uno	las	una	tu	yo
5	4	4		el	la	los	uno	las	una	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8

swap	i	j	a							
			1	2	3	4	5	6	7	8
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo
	4	2	el	la	los	uno	las	una	tu	yo
	5	4	el	la	los	uno	los	una	tu	yo
	5	3	la	la	los	los	las	uno	una	tu

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8

swap	i	j	a							
			1	2	3	4	5	6	7	8
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo
	4	2	el	la	los	uno	las	una	tu	yo
	5	4	el	la	los	uno	los	una	tu	yo
	5	3	la	la	los	los	las	uno	una	tu
	5	2	el	la	las	los	uno	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8

swap	i	j	a							
			1	2	3	4	5	6	7	8
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo
	4	2	el	la	los	uno	las	una	tu	yo
	5	4	el	la	los	uno	las	una	tu	yo
	5	3	la	la	los	las	uno	una	tu	yo
	5	2	el	la	las	los	uno	una	tu	yo
	6	5	el	la	las	los	uno	una	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8

swap	i	j	a							
			1	2	3	4	5	6	7	8
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo
	4	2	el	la	los	uno	las	una	tu	yo
	5	4	el	la	los	uno	las	una	tu	yo
	5	3	la	la	los	las	uno	una	tu	yo
	5	2	el	la	las	los	uno	una	tu	yo
	6	5	el	la	las	los	uno	una	tu	yo
	6	4	el	la	las	los	una	uno	tu	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8

swap	i	j	a							
			1	2	3	4	5	6	7	8
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo
	4	2	el	la	los	uno	las	una	tu	yo
	5	4	el	la	los	uno	las	una	tu	yo
	5	3	la	la	los	las	uno	una	tu	yo
	5	2	el	la	las	los	uno	una	tu	yo
	6	5	el	la	las	los	uno	una	tu	yo
	6	4	el	la	las	los	una	uno	tu	yo
	7	6	el	la	las	los	una	uno	tu	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8

swap	i	j	a							
			1	2	3	4	5	6	7	8
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo
	4	2	el	la	los	uno	las	una	tu	yo
	5	4	el	la	los	uno	las	una	tu	yo
	5	3	la	la	los	las	uno	una	tu	yo
	5	2	el	la	las	los	uno	una	tu	yo
	6	5	el	la	las	los	uno	una	tu	yo
	6	4	el	la	las	los	una	uno	tu	yo
	7	6	el	la	las	los	una	uno	tu	yo
	7	5	el	la	las	los	una	tu	uno	yo

insertionsort(A):

Para i de 2 a N :

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8

swap	i	j	a							
			1	2	3	4	5	6	7	8
	2	1	la	el	uno	los	las	una	tu	yo
	3	2	el	la	uno	las	las	una	tu	yo
	4	3	el	la	uno	los	las	una	tu	yo
	4	2	el	la	los	uno	las	una	tu	yo
	5	4	el	la	los	uno	las	una	tu	yo
	5	3	la	la	los	las	uno	una	tu	yo
	5	2	el	la	las	los	uno	una	tu	yo
	6	5	el	la	las	los	uno	una	tu	yo
	6	4	el	la	las	los	una	uno	tu	yo
	7	6	el	la	las	los	una	uno	tu	yo
	7	5	el	la	las	los	una	tu	uno	yo
	7	4	el	la	las	los	tu	una	uno	yo

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

$$j = j - 1$$

$$A_{j+1} = k$$

Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Ordenamiento por Inserción

- **Iteración i :** intercambiar repetidamente a_i si el elemento a la izquierda es menor
- **Propiedad:** después de i -ésima iteración, los elementos entre a_0 y a_i están ordenados.

el	la	las	los	tu	una	uno	yo
1	2	3	4	5	6	7	8

swap

	i	j	a	1	2	3	4	5	6	7	8
2	1	1	la	el	uno	los	las	una	tu	yo	
3	2	2	el	la	uno	las	las	una	tu	yo	
4	3	3	el	la	uno	los	las	una	tu	yo	
4	2	2	el	la	los	uno	las	una	tu	yo	
5	4	4	el	la	los	uno	las	una	tu	yo	
5	3	3	la	la	los	las	uno	una	tu	yo	
5	2	2	el	la	las	los	uno	una	tu	yo	
6	5	5	el	la	las	los	uno	una	tu	yo	
6	4	4	el	la	las	los	una	uno	tu	yo	
7	6	6	el	la	las	los	una	uno	tu	yo	
7	5	5	el	la	las	los	una	tu	uno	yo	
7	4	4	el	la	las	los	tu	una	uno	yo	
8	7	7	el	la	las	los	tu	una	uno	yo	

insertionsort(A):

Para i de 2 a N:

$$k = A_i$$

$$j = i - 1$$

Mientras $j \geq 1$ and $A_j > k$:

$$A_{j+1} = A_j$$

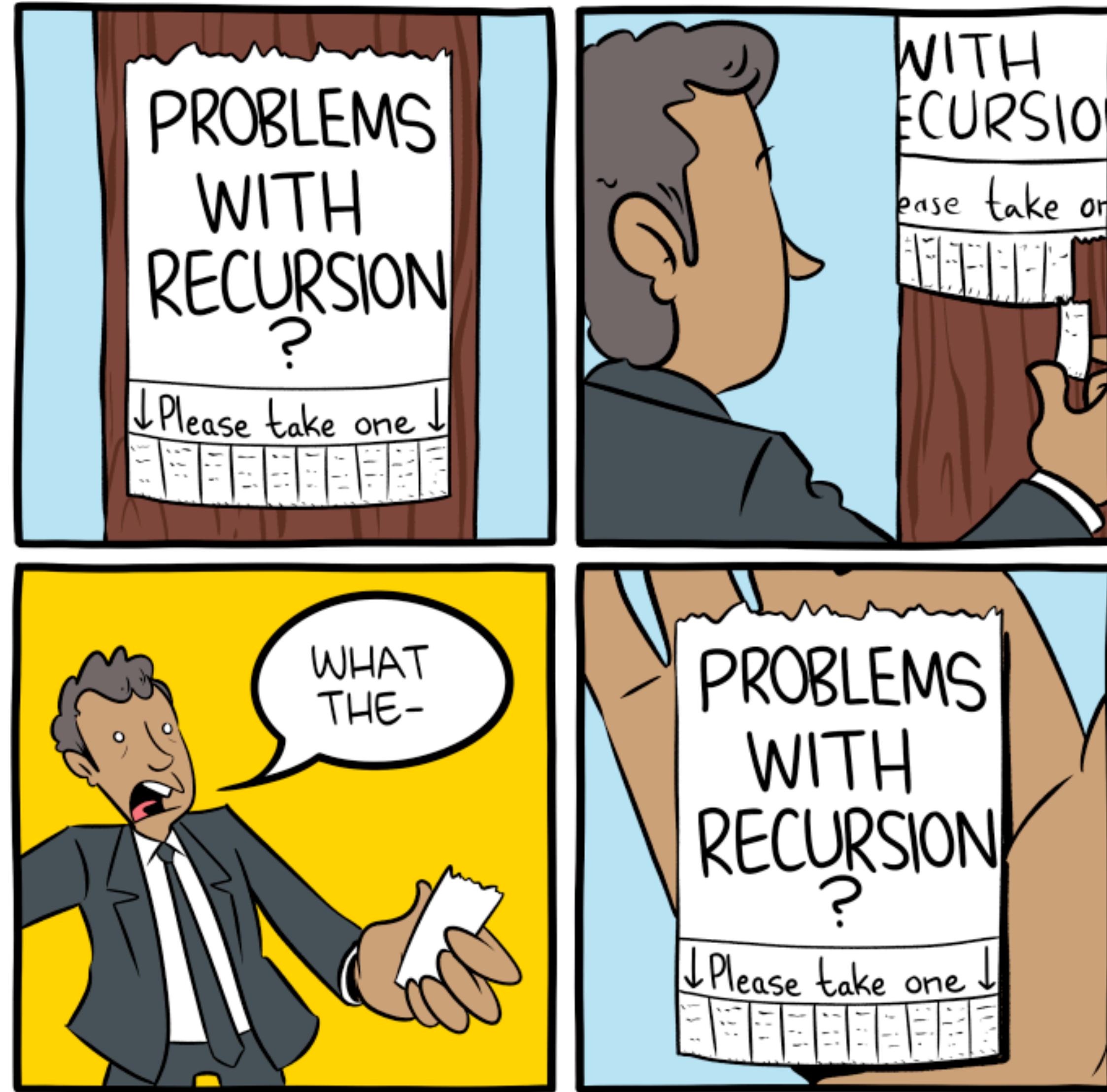
$$j = j - 1$$

$$A_{j+1} = k$$

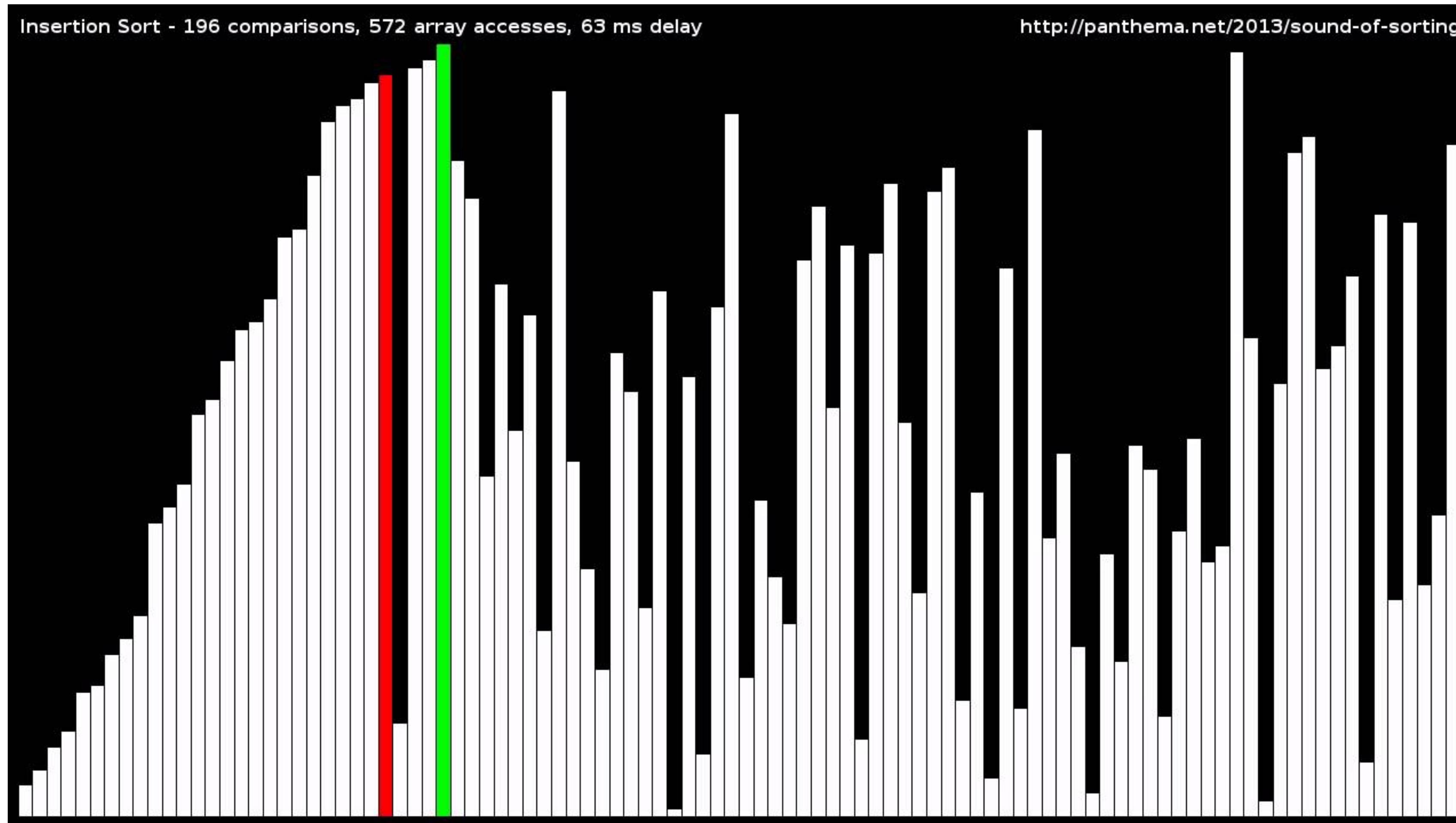
Idea: Insertar A_i en la sublista ordenada A_1 hasta A_j

Actividad

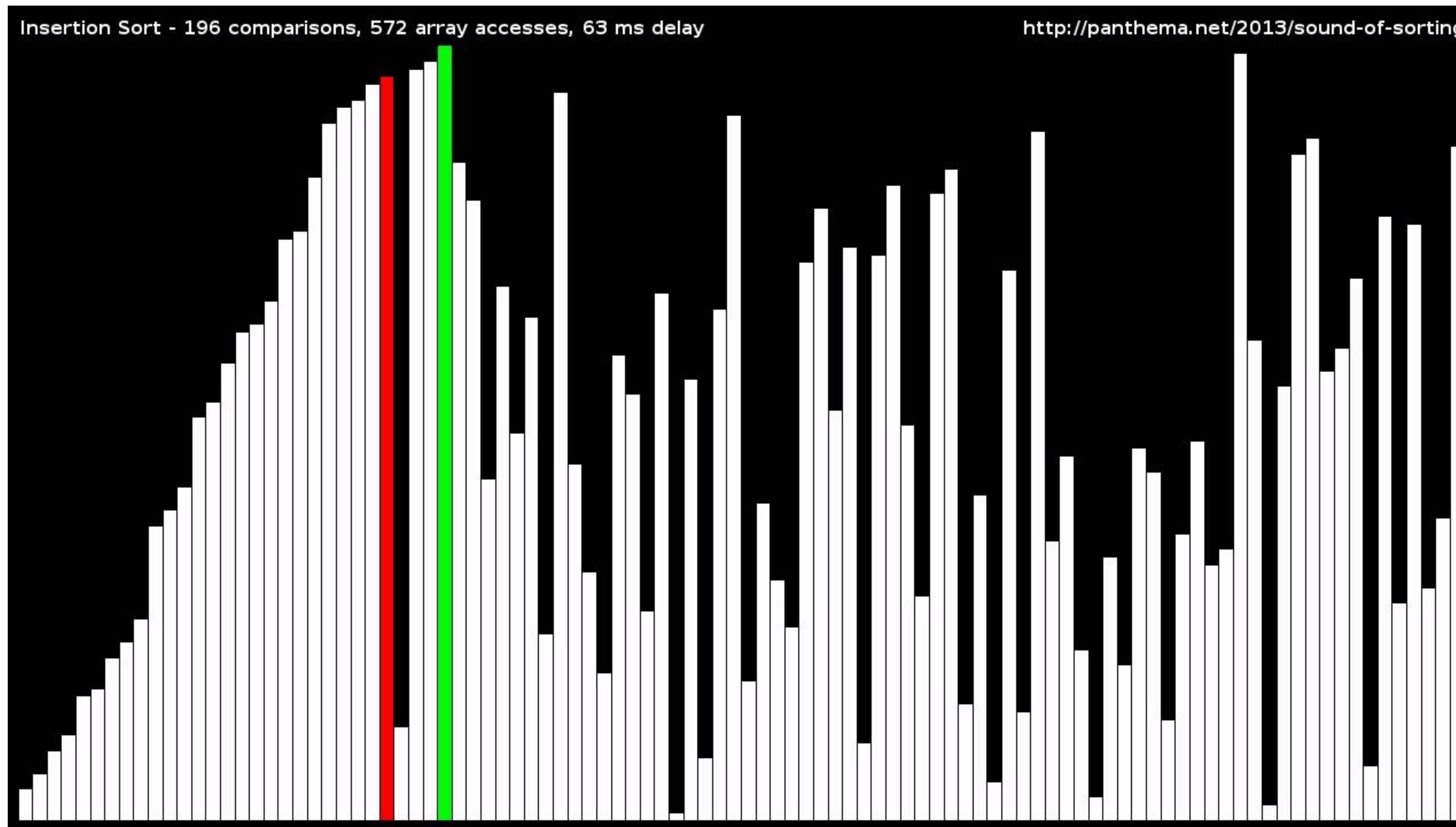
- Hacer la traza del algoritmo insertion sort
- Haga el diagrama de flujo del algoritmo insertionsort
- Cuántas comparaciones $A_j > k$ (en el ciclo Mientras) se ejecutan para estas 3 entradas:
 1. $A = 5,4,3,2,1$
 2. $A = 1,2,3,4,5$
 3. $A = 1,2,5,4,3$
- Con respecto a la pregunta anterior, ¿cuál diría que es el “mejor caso” y “peor caso” del algoritmo? (entendiendo que entre más comparaciones se realizan, más tiempo tomará ejecutar el algoritmo)
- Si el tamaño de la lista A es N, cuántas comparaciones se realizan en el mejor caso? y en el peor caso?



[smbc-comics.com](https://www.smbc-comics.com/comic/recursion)



<https://www.youtube.com/watch?v=8oJS1BMKE64>



<https://www.youtube.com/watch?v=8oJS1BMKE64>

insertionsort(A):

Para i de 2 a N-1:

k = A_i

j = i - 1

Mientras j >= 1 and A_j > k:

A_{j+1} = A_j

j = j - 1

A_{j+1} = k