

SESIÓN /03

OPERACIONES CON VECTORES

/ ORDENAMIENTO

/ INSERCIÓN

/ INTRODUCCIÓN

- La clase anterior conocimos las principales operaciones sobre un Arreglo unidimensional, conocido también como Vector.
- En la presente sesión, conoceremos los principales algoritmos para realizar las operaciones de búsqueda y de ordenamiento sobre un Vector.
- Culminaremos analizando la eficiencia de estos algoritmos a fin de seleccionar la mejor opción.

ORDENAMIENTO

/ ORDENAMIENTO

Definición

Se trata de establecer un orden (ascendente ó descendente) entre los elementos del Vector.

Ejemplo: En el siguiente Vector, ordena sus elementos

2	A[0]
9	A[1]
4	A[2]
7	A[3]
1	A[4]

/ ORDENAMIENTO

Definición

Se trata de establecer un orden (ascendente ó descendente) entre los elementos del Vector.

Ejemplo: En el siguiente Vector, ordena sus elementos

2	A[0]
9	A[1]
4	A[2]
7	A[3]
1	A[4]

Resultado:

1	A[0]
2	A[1]
4	A[2]
7	A[3]
9	A[4]

/ ORDENAMIENTO

Algoritmos

- Ordenamiento Burbuja
- Ordenamiento Quick Sort
- Ordenamiento Shell

ORDENAMIENTO BURBUJA

/ ORDENAMIENTO BURBUJA

Descripción

Para implementar este algoritmo es necesario recorrer el Vector varias veces.

El proceso de recorrer el Vector se llama rueda ó ronda.

Así, el ordenamiento se realiza en varias **ruedas de comparaciones**.

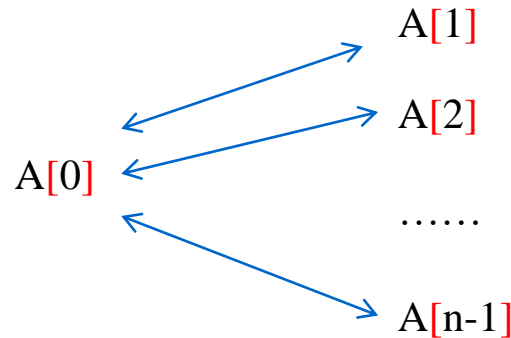
A	0	1	n-2	n-1
	5	2	7	3

/ ORDENAMIENTO BURBUJA

Descripción

Rueda 1

Se compara el primer elemento del Vector con los demás elementos. Cada vez que se encuentra un elemento menor al elemento de la posición 0, se intercambian sus valores.



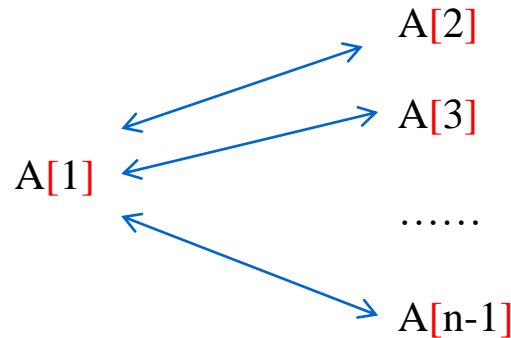
Número de comparaciones: $n - 1$

/ ORDENAMIENTO BURBUJA

Descripción

Rueda 2

Se compara el segundo elemento del Vector con los demás elementos. Cada vez que se encuentra un elemento menor al elemento de la posición 1, se intercambian sus valores.



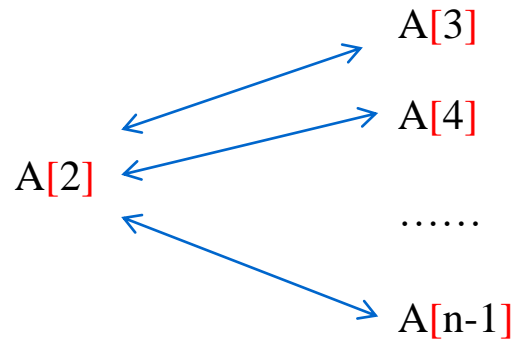
Número de comparaciones: $n - 2$

/ ORDENAMIENTO BURBUJA

Descripción

Rueda 3

Se compara el tercer elemento del Vector con los demás elementos. Cada vez que se encuentra un elemento menor al elemento de la posición 2, se intercambian sus valores.



Número de comparaciones: $n - 3$

/ ORDENAMIENTO BURBUJA

Descripción

Rueda $n - 1$

Se compara el penúltimo elemento con el último elemento del Vector. Si el penúltimo elemento es mayor al último elemento, se intercambian sus valores.

$$A[n-2] \longleftrightarrow A[n-1]$$

Número de comparaciones: 1

/ ORDENAMIENTO BURBUJA

Ejemplo

Ordena el siguiente Vector:

A	0	1	2	3	4
	20	90	40	70	10

El Vector tiene 5 elementos, por lo tanto, se requiere 4 ruedas de comparaciones para ordenarlo.

/ ORDENAMIENTO BURBUJA

Ejemplo

Rueda 1

0	1	2	3	4
20	90	40	70	10
20	90	40	70	10
20	90	40	70	10
20	90	40	70	10
10	90	40	70	20

¿ $A[0] > A[1]$?

¿ $A[0] > A[2]$?

¿ $A[0] > A[3]$?

¿ $A[0] > A[4]$?

Número de comparaciones: 4

Número de intercambios: 1

/ ORDENAMIENTO BURBUJA

Ejemplo

Rueda 2

0	1	2	3	4
10	90	40	70	20

¿ $A[1] > A[2]$?

10	40	90	70	20
----	----	----	----	----

¿ $A[1] > A[3]$?

10	40	90	70	20
----	----	----	----	----

¿ $A[1] > A[4]$?

10	20	90	70	40
----	----	----	----	----

Número de comparaciones: 3

Número de intercambios: 2

/ ORDENAMIENTO BURBUJA

Ejemplo

Rueda 3

0	1	2	3	4
10	20	90	70	40
10	20	70	90	40
10	20	40	90	70

¿ $A[2] > A[3]$?

¿ $A[2] > A[4]$?

Número de comparaciones: 2

Número de intercambios: 2

/ ORDENAMIENTO BURBUJA

Ejemplo

Rueda 4

0	1	2	3	4
10	20	40	90	70
10	20	40	70	90

¿ $A[3] > A[4]$?

Número de comparaciones: 1

Número de intercambios: 1

/ ORDENAMIENTO BURBUJA

Eficiencia

En la primera ronda se realizan $n-1$ comparaciones, en la segunda ronda se realizan $n-2$ comparaciones y así sucesivamente.

Entonces, el número total de comparaciones será:

$$(n-1) + (n-2) + (n-3) + \dots + 3 + 2 + 1$$

La fórmula para determinar la suma de esta progresión es:

$$\frac{n \times (n - 1)}{2}$$

Para un Vector que tiene n elementos

INSERCIÓN

/ INSERCIÓN

Definición

Consiste en agregar un elemento en una determinada posición.

La inserción puede ser:

- Al inicio.
- Al final (adicionar).
- Entre dos elementos.

/ INSERCIÓN

Ejemplo

En el siguiente Vector, inserta el número 8 en la posición 1

2	A[0]
4	A[1]
7	A[2]
1	A[3]
	A[4]

/ INSERCIÓN

Ejemplo

En el siguiente Vector, inserta el número 8 en la posición 1

2	A[0]
4	A[1]
7	A[2]
1	A[3]
	A[4]

Resultado:

2	A[0]
8	A[1]
4	A[2]
7	A[3]
1	A[4]

*Se realizaron 3
traslados*

/ CONCLUSIONES

- El ordenamiento de los elementos de un Vector puede ser en forma ascendente ó descendente. Si no se especifica, se asume la primera forma.
- La eficiencia en los algoritmos de ordenamiento se mide por el número de comparaciones a realizar.
- La eficiencia en los algoritmos de inserción y eliminación se mide por el número de traslados a realizar.

/ BIBLIOGRAFÍA

- Cairo, O.; Guardati, S. (2008). Estructuras de datos. 3ra. Edición. México D.F., Mexico: McGraw Hill.
- Instituto NIIT (2011). Data Structures and Algorithms. Student guide.