UNIVERSIDAD DE CORDOBA

FACULTAD DE INGENIERIAS

PROGRAMA INGENIERIA DE SISTEMAS

A SIGNATURA PROGRAMACION II

DESCRIPCION METODO DE INTERCAMBIO

PROFESOR LUIS ROBERTO OLASCOAGA

MONTERIA SEPTIEMBRE DE 2019

1. Método de ordenamiento por intercambio

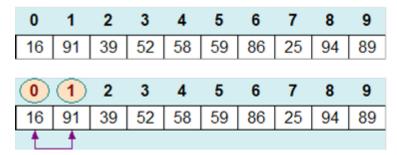
El método de intercambio es uno de los más sencillos de comprender e implementar, pero así mismo uno de los menos eficientes va que tiene que comparar cada elemento del arreglo con aquellos que se encuentra a su derecha, generándose en consecuencia un alto número de comparaciones sin contar los posibles intercambios de elementos que sucedan que cada caso. Entrando en materia, el algoritmo para ordenar un arreglo por intercambio consiste en comparar cada elemento del arreglo con todos los que le siguen; es decir, con aquellos que ocupan posiciones mayores que él. En otras palabras este método toma cada elemento del arregio y lo compara con todos aquellos elementos que se encuentran su derecha, por lo que decimos que las parejas de elementos comparados por el método de intercambio son de la forma (i. i + k) donde k es un número mayor o igual que uno y por supuesto el elemento de la expresión i+k debe ser menor o igual a elemento de la posición mayor del arreglo. De esta manera con cada para de elementos comparados, si dichos elementos no están en el orden adecuado entonces se procede a intercambiarlos de posición. Para el caso de un ordenamiento ascendente; es decir, de menor a mayor, el intercambio se efectuara siempre que el elemento de la izquierda (posición i) sea mayor que el de su derecha (posición i+k).

En este orden de ideas a para un ordenamiento ascendente, el método de ordenamiento por intercambio empieza comparando el primer elemento del arreglo (el que ocupa la posición cero) con todos los que le siguen; es decir, con todos aquellos comprendidos desde la posición uno hasta la última posición valida dentro del arreglo (en general la N-1 siendo N el tamaño del arreglo), haciendo intercambios en cada comparación cuando los elementos comparados no estén en el orden adecuado. Al final de todas las comparaciones hechas con el primer elemento resulta que queda un valor correctamente acomodado en el arreglo, el cual ocupa la primera posición de este; es decir, el puesto cero. Este primer elemento ordenado será el valor más pequeño del arreglo si estamos ordenado de menor a mayor, o el valor más grande del mismo si estamos ordenando de mayor a menor. Para ubicar en la posición correcta al resto de elementos del arreglo, el proceso se repite con la misma dinámica, pero esta vez iniciando con el segundo

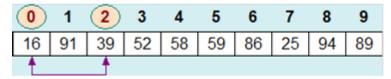
elemento del arreglo, comparando a dicho elemento con todos los que le siguen, desde la posición del tercer elemento hasta el último. El proceso se repite comparando los elementos siguientes con todos los que están a su derecha, hasta que en la última pasada solo se comparan los elementos de la penúltima y de la última posición del arreglo, momento en que ya queda ordenado el arreglo.

2. Representación del método de intercambio.

Como ejemplo consideraremos el siguiente vector de diez números enteros:



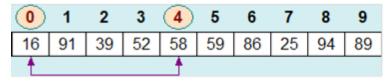
Comparamos las posiciones 0 y 1: Vec[0]>Vec[1]-> es 16>91? no.



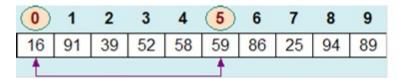
Comparamos las posiciones 0 y 2: Vec[0]>Vec[2] -> es 16>39? no.

0	1	2	3	4	5	6	7	8	9
16	91	39	52	58	59	86	25	94	89
<u> †</u> _									

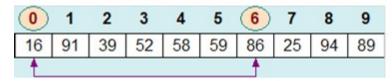
Comparamos las posiciones 0 y 3: Vec[0]>Vec[3] -> es 16>52? no.



Comparamos las posiciones 0 y 4: Vec[0]>Vec[4] -> es 16>58? no.



Comparamos las posiciones 0 y 5: Vec[0]>Vec[5] -> es 16>59? no.



Comparamos las posiciones 0 y 6: Vec[0]>Vec[6] -> es 16>86? no.

0	1	2	3	4	5	6	7	8	9
16	91	39	52	58	59	86	25	94	89
*							1		

Comparamos las posiciones 0 y 7: Vec[0]>Vec[7] -> es 16>25? no.

0	1	2	3	4	5	6	7	8	9
16	91	39	52	58	59	86	25	94	89
•			200					1	

Comparamos las posiciones 0 y 8: Vec[0]>Vec[8] -> es 16>94? no.

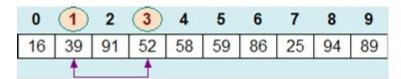
		2							
16	91	39	52	58	59	86	25	94	89
4									+

Comparamos las posiciones 0 y 9: Vec[0]>Vec[9] -> es 16>89? no.

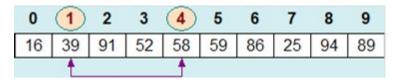
0	1	2	3	4	5	6	7	8	9
16	91	39	52	58	59	86	25	94	89
	1	*							

Comparamos las posiciones 1 y 2: Vec[1]>Vec[2] -> es 91>39? si, intercambiamos las posiciones 1 y 2 quedando:

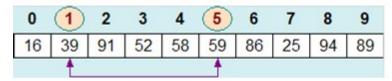
	1	3770	1000	1000	- 2			1000	
16	39	91	52	58	59	86	25	94	89



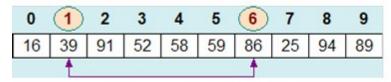
Comparamos las posiciones 1 y 3: Vec[1]>Vec[3] -> es 39>52? no.



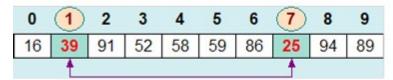
Comparamos las posiciones 1 y 4: Vec[1]>Vec[4] -> es 39>58? no.



Comparamos las posiciones 1 y 5: Vec[1]>Vec[5] -> es 39>59? no.



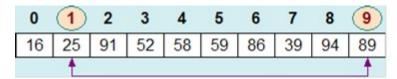
Comparamos las posiciones 1 y 6: Vec[1]>Vec[6] -> es 39>86? no.



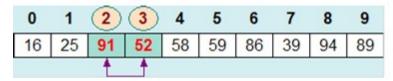
Comparamos las posiciones 1 y 7: Vec[1]>Vec[7] -> es 39>25? si, intercambiamos las posiciones 1 y 7 quedando:



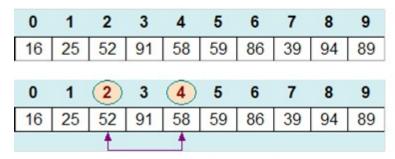
Comparamos las posiciones 1 y 8: Vec[1]>Vec[8] -> es 25>94? no.



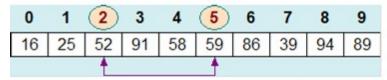
Comparamos las posiciones 1 y 9: Vec[1]>Vec[9] -> es 25>89? no.



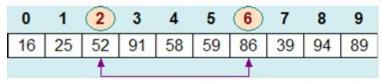
Comparamos las posiciones 2 y 3: Vec[2]>Vec[3] -> es 91>52? si, intercambiamos las posiciones 2 y 3 quedando:



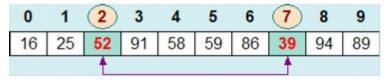
Comparamos las posiciones 2 y 4: Vec[2]>Vec[4] -> es 52>58? no.



Comparamos las posiciones 2 y 5: Vec[2]>Vec[5] -> es 52>59? no.



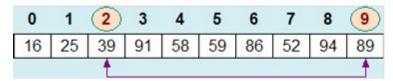
Comparamos las posiciones 2 y 6: Vec[2]>Vec[6] -> es 52>86? no.



Comparamos las posiciones 2 y 7: Vec[2]>Vec[7] -> es 52>39? si, intercambiamos las posiciones 2 y 7 quedando:



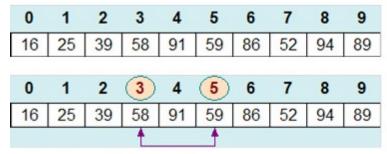
Comparamos las posiciones 2 y 8: Vec[2]>Vec[8] -> es 39>94? no.



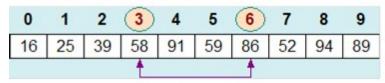
Comparamos las posiciones 2 y 9: Vec[2]>Vec[9] -> es 39>89? no.



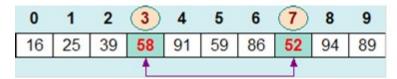
Comparamos las posiciones 3 y 4: Vec[3]>Vec[4] -> es 91>58? si, intercambiamos las posiciones 3 y 4 quedando:



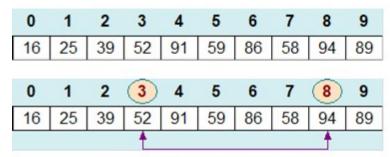
Comparamos las posiciones 3 y 5: Vec[3]>Vec[5] -> es 58>59? no.



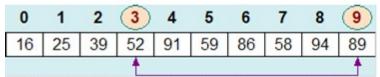
Comparamos las posiciones 3 y 6: Vec[3]>Vec[6] -> es 58>86? no.



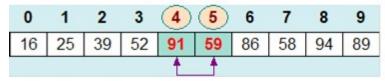
Comparamos las posiciones 3 y 7: Vec[3]>Vec[7] -> es 58>52? si, intercambiamos las posiciones 3 y 7 quedando:



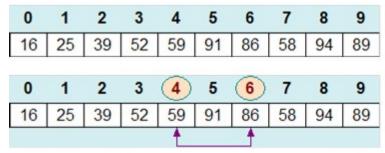
Comparamos las posiciones 3 y 8: Vec[3]>Vec[8] -> es 52>94? no.



Comparamos las posiciones 3 y 9: Vec[3]>Vec[9] -> es 52>89? no.



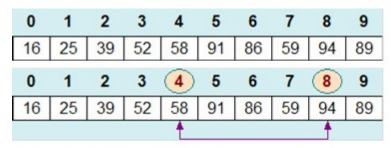
Comparamos las posiciones 4 y 5: Vec[4]>Vec[5] -> es 91>59? si, intercambiamos las posiciones 4 y 5 quedando:



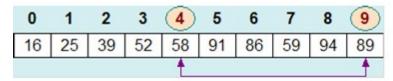
Comparamos las posiciones 4 y 6: Vec[4]>Vec[6] -> es 59>86? no.

0	1	2	3	4	5	6	7	8	9
16	25	39	52	59	91	86	58	94	89
				1			1		

Comparamos las posiciones 4 y 7: Vec[4]>Vec[7] -> es 59>58? si, intercambiamos las posiciones 4 y 7 quedando:



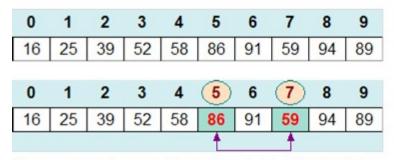
Comparamos las posiciones 4 y 8: Vec[4]>Vec[8] -> es 58>94? no.



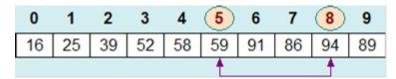
Comparamos las posiciones 4 y 9: Vec[4]>Vec[9] -> es 58>89? no.

0	1	2	3	4	5	6	7	8	9
16	25	39	52	58	91	86	59	94	89
					<u> </u>	•			

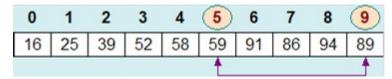
Comparamos las posiciones 5 y 6: Vec[5]>Vec[6] -> es 91>86? si, intercambiamos las posiciones 5 y 6 quedando:



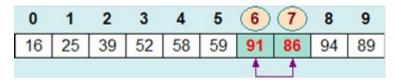
Comparamos las posiciones 5 y 7: Vec[5]>Vec[7] -> es 86>59? si, intercambiamos las posiciones 5 y 7 quedando:



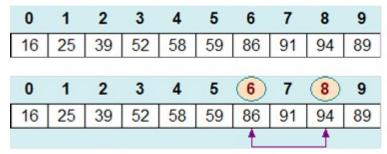
Comparamos las posiciones 5 y 8: Vec[5]>Vec[8] -> es 59>94? no.



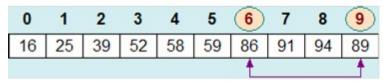
Comparamos las posiciones 5 y 9: Vec[5]>Vec[9] -> es 59>89? no.



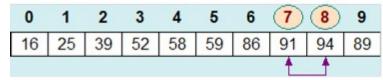
Comparamos las posiciones 6 y 7: Vec[6]>Vec[7] -> es 91>86? si, intercambiamos las posiciones 6 y 7 quedando:



Comparamos las posiciones 6 y 8: Vec[6]>Vec[8] -> es 86>94? no.



Comparamos las posiciones 6 y 9: Vec[6]>Vec[9] -> es 86>89? no.



Comparamos las posiciones 7 y 8: Vec[7]>Vec[8] -> es 91>94? no.

0	1	2	3	4	5	6	7	8	9
16	25	39	52	58	59	86	91	94	89
		-					1		+

Comparamos las posiciones 7 y 9: Vec[7]>Vec[9] -> es 91>89? si, intercambiamos las posiciones 7 y 9 quedando:

0	1	2	3	4	5	6	7	8	9
16	25	39	52	58	59	86	89	94	91
								_	_
0	1	2	3	4	5	6	7	8	9
0	1 25	39	3 52	4 58	5	6	7	94	91

Comparamos las posiciones 8 y 9: Vec[8]>Vec[9] -> es 94>91? si, intercambiamos las posiciones 8 y 9 quedando:

	1	_	-	_	_		-	_	
16	25	39	52	58	59	86	89	91	94