

CLASE 4

Ing. Silvestre Alejandro
Informática III
IUA - 2024



Objetivos del día

1. CodeReview
2. Ordenamiento
3. Presentación segundo proyecto



CodeReview



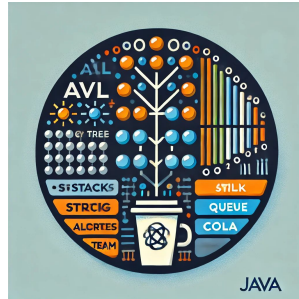


CodeReview

1. Introducción de que el proyecto.
2. Explicar y mostrar estructura general de carpetas, clases, etc.
3. Explicar y mostrar el código de al menos un ejercicio.
4. Demo de la funcionalidad.



Ordenamiento





Inserción

Consiste en recorrer todo el array comenzando desde el segundo elemento hasta el final. Para cada elemento, se trata de colocarlo en el lugar correcto entre todos los elementos anteriores a él o sea entre los elementos a su izquierda en el array.

30	15	2	21	44	8	Array original
30	15	2	21	44	8	Se empieza por el segundo elemento. Se compara con el primero. Como $15 < 30$ se desplaza el 30 hacia la derecha y se coloca el 15 en su lugar
15	30	2	21	44	8	
15	30	2	21	44	8	Seguimos por el tercer elemento. Se compara con los anteriores y se van desplazando hasta que el 2 queda en su lugar.
2	15	30	21	44	8	
2	15	30	21	44	8	Continuamos por el cuarto elemento. Se compara con los anteriores y se van desplazando hasta que el 21 queda en su lugar.
2	15	21	30	44	8	
2	15	21	30	44	8	Lo mismo para el quinto elemento En este caso ya está en su posición correcta respecto a los anteriores.
2	15	21	30	44	8	
2	15	21	30	44	8	Y finalmente se coloca el último elemento El array queda ordenado
2	8	15	21	30	44	



Shellsort

Se van dando pasadas con el mismo salto hasta que en una pasada no se intercambie ningún elemento de sitio. Entonces el salto se reduce a la mitad, y se vuelven a dar pasadas hasta que no se intercambie ningún elemento, y así sucesivamente hasta que el salto vale 1.

50	26	7	9	15	27	Array original
Salto = 3						
9	26	7	50	15	27	Se intercambian 9 y 50
9	15	7	50	26	27	Se intercambian 15 y 26
Salto = 1						
9	7	15	50	26	27	Se intercambian 15 y 7
9	7	15	26	50	27	Se intercambian 50 y 26
9	7	15	26	27	50	Se intercambian 50 y 27
Segunda Pasada con Salto 1:						
7	9	15	26	27	50	Se intercambian 9 y 7, Array ordenado



Quicksort

Se basa en la técnica divide y vencerás, que consiste en ir subdividiendo el array en arrays más pequeños, y ordenar éstos.

10	40	7	9	15	27	Array original a ordenar
10	40	7	9	15	27	Se toma como pivote el primer elemento
10	40	7	9	15	27	La búsqueda de izquierda a derecha encuentra el 40, mayor que pivote y la búsqueda de derecha a izquierda encuentra el 9, menor que pivote
10	9	7	40	15	27	Se intercambian
10	9	7	40	15	27	Continúa la búsqueda, se encuentra el valor 40 mayor que pivote y el valor 7 menor que pivote, pero ya se han cruzado. Paramos.
7	9	10	40	15	27	Finalmente colocamos el pivote en su lugar. (Posición j), quedando el array dividido en dos subarrays a los que les aplicará el mismo proceso
subarray 1		subarray 2				

¡Vamos a al práctico 2!



¡Gracias!

