

Ayudantía 7

ESTRUCTURA DE DATOS Y ALGORITMOS

PROFESOR: YERKO ORTIZ

AYUDANTE: VICENTE DÍAZ

SECCIÓN 6

Contacto

Discord: trapitokid

Wsp: +569 5749 6014

LoL: Saki TvT#woof





Sequential search

steps: 0

37

1	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Búsqueda Binaria



Binary search

steps: 0

37

1	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Low								mid								high

Búsqueda Binaria con Conteo de Pasos



Se tiene un arreglo de números enteros ordenado de forma ascendente. Tu tarea es implementar una función llamada `BinarySearch` que busque un número específico (target) dentro del arreglo utilizando el algoritmo de **búsqueda binaria**.

Entrada:

- Un número entero `n` que representa el tamaño del arreglo.
- Una secuencia de `n` números enteros ordenados en forma ascendente.
- Un número entero `target`, que es el número que se desea buscar en el arreglo.

Salida:

- Si el número se encuentra en el arreglo:
 - Imprime el índice donde se encuentra (comenzando desde 0).
 - Imprime la cantidad de pasos (iteraciones) que tomó encontrar el número.
- Si el número no se encuentra en el arreglo:
 - Imprime el mensaje "Número no encontrado".

Comparadores y .sort

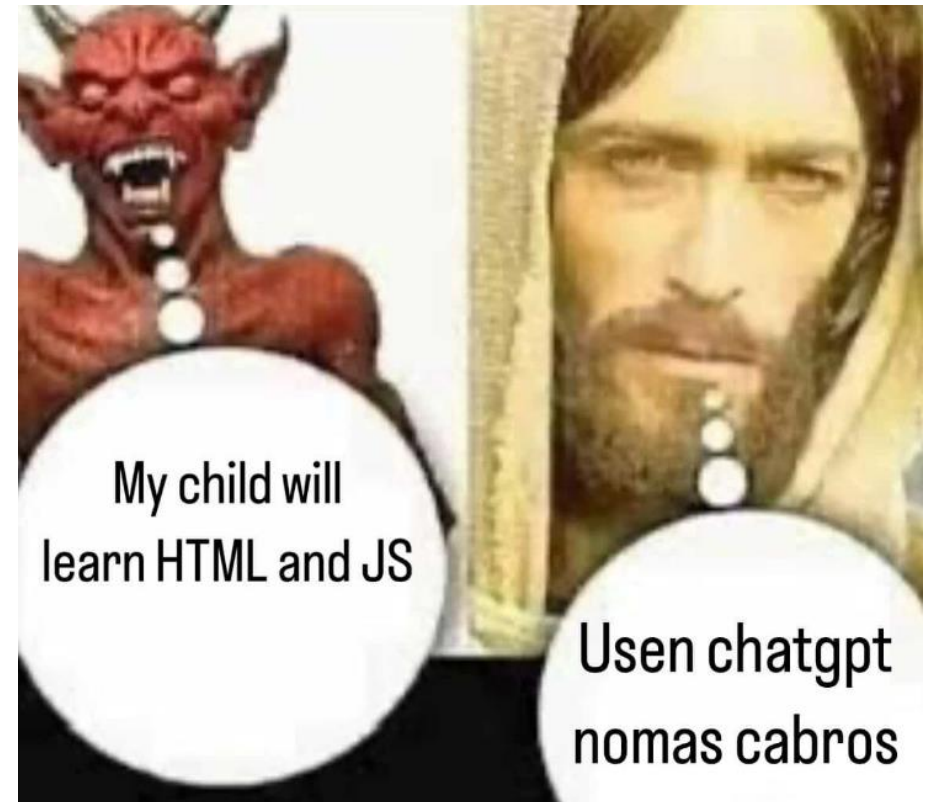
Comparator:

- Es una función que compara 2 objetos a y b, puedes tener 3 valores:
 - < 0 : a es menor que b
 - 0 : a es igual que b
 - > 0 : a es mayor que b
- Utilizaremos 2 tipos de comparators:
 - `Integer.compare(a, b)`
 - `Double.compare(a, b)`

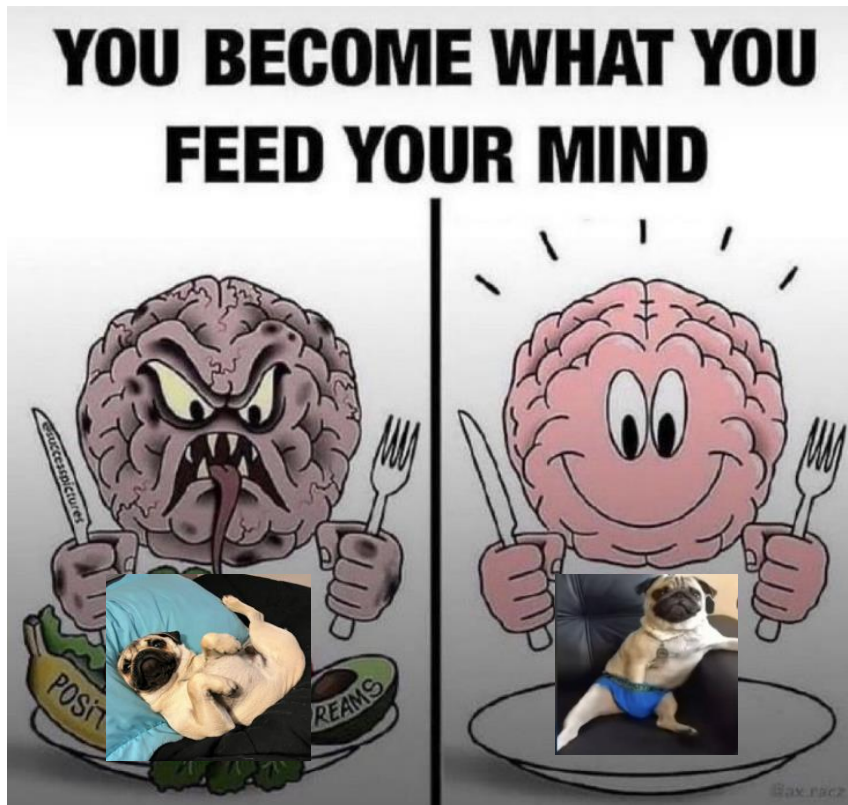
`Array.sort()`:

- Método de la clase `Arrays` en `java.util`
- Permite ordenar arreglos de objetos
- Utiliza un comparador personalizado que indica cómo comparar los elementos.

`Arrays.sort(array[] arr, (a,b) -> Comparator)`



Ordenar puntos para análisis geométrico



Una empresa de análisis geométrico necesita trabajar con una colección de puntos en el plano cartesiano para realizar cálculos como identificar los puntos más cercanos al origen, ordenar coordenadas por eje, o incluso comparar ubicaciones.

Tienes una clase Point que representa un punto en 2D con coordenadas x e y. Cada punto es generado de forma aleatoria, con valores enteros entre 0 y 9.

Se debe:

- Ordenarlos utilizando la librería estándar de Java (Arrays.sort) de acuerdo a distintos criterios:
 - a) Por coordenada x.
 - b) Por coordenada y.
 - c) Por distancia desde el origen (0,0) usando la fórmula: $d = \sqrt{x^2 + y^2}$
- Mostrar los puntos antes y después de cada ordenamiento, para visualizar el efecto de cada criterio.