#1

Radix Sort

3 DE JUNIO DEL 2021

INTEGRANTES

GRUPO #1

Elías Robles Montero - B96551
Alejandra Camacho Ochoa - B91470
José Salas Calderón - C07051

INFORMACIÓN GENERAL

Nos enfocaremos en:

- Radix Sort para ordenar números enteros, sin embargo, no se limita solamente a eso, se pueden ordenar otros tipos como String.

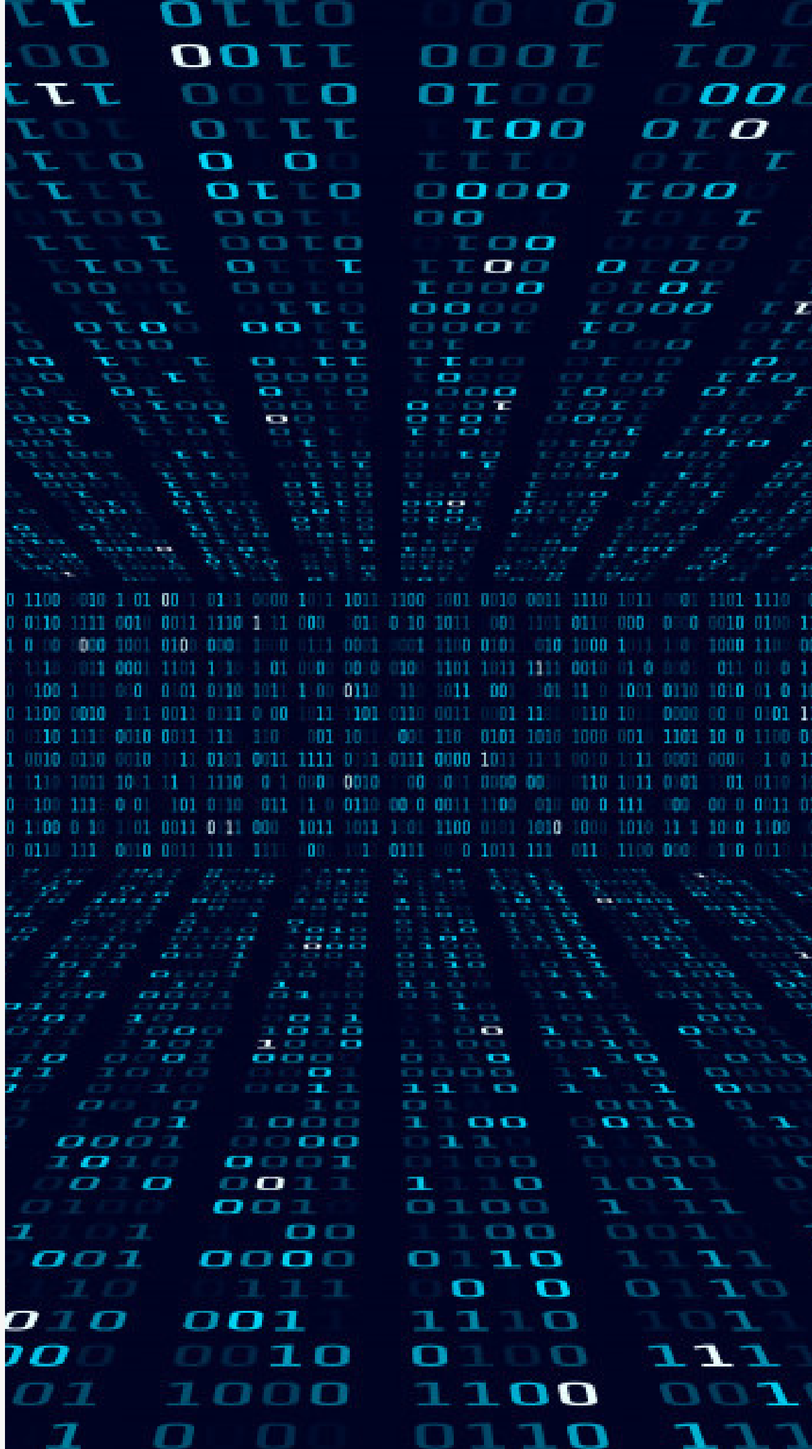
DESCRIPCIÓN GENERAL DEL ALGORITMO

Posiciones.

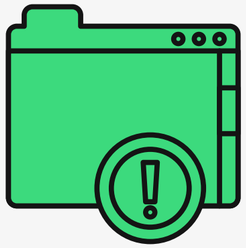
La ordenación por radix es un algoritmo de clasificación que ordena los números en función de las posiciones de sus dígitos.

No compara.

A diferencia de la mayoría de los otros algoritmos de ordenación, como Merge Sort, Insertion Sort, Bubble Sort, no compara los números.



UN EJEMPLO RÁPIDO:

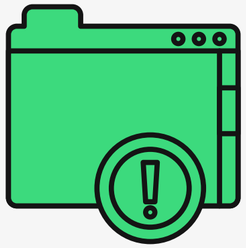


- Consideremos el siguiente arreglo:

387	468	134	123	68	221	769	37	7

Imagenes tomadas desde:
https://www.baeldung.com/java-radix-sort#:~:text=Radix%20sort%20is%20a%20sorting,doesn't%20compare%20the%20numbers

UN EJEMPLO RÁPIDO:



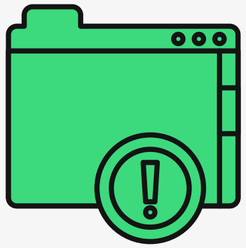
- Consideremos el siguiente arreglo:

387	468	134	123	68	221	769	37	7

- Se enfocará en los primeros digitos:

387	468	134	123	68	221	769	37	7

UN EJEMPLO RÁPIDO:



- Consideremos el siguiente arreglo:

387	468	134	123	68	221	769	37	7

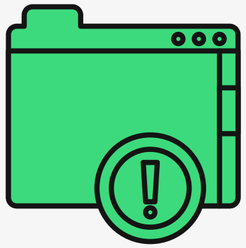
- Se enfocará en los primeros digitos:

387	468	134	123	68	221	769	37	7

- Después de la primera iteración:

221	123	134	387	37	7	468	68	769

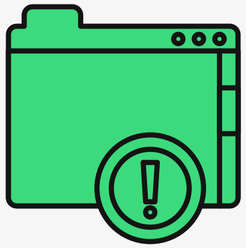
UN EJEMPLO RÁPIDO:



- Ahora el algoritmo evaluará los segundos dígitos:

221	123	134	387	37	7	468	68	769

UN EJEMPLO RÁPIDO:



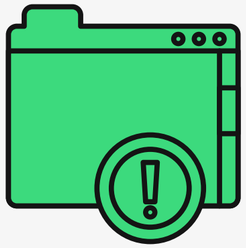
- Ahora el algoritmo evaluará los segundos dígitos:

221	123	134	387	37	7	468	68	769

- Y se mostrarían los arreglos de la siguiente manera:

7	221	123	134	37	468	68	769	387

UN EJEMPLO RÁPIDO:



- Ahora el algoritmo evaluará los segundos dígitos:

221	123	134	387	37	7	468	68	769

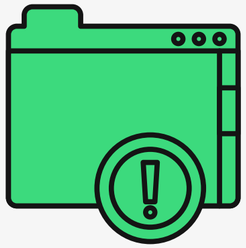
- Y se mostrarían los arreglos de la siguiente manera:

7	221	123	134	37	468	68	769	387

- Para finalizar, se enfocaría en la tercera posición de dígitos:

7	221	123	134	37	468	68	769	387

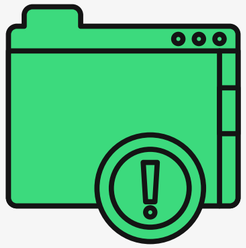
UN EJEMPLO RÁPIDO:



- Y la salida final del algoritmo seria la siguiente:

7	37	68	123	134	221	387	468	769

UN EJEMPLO RÁPIDO:



- Y la salida final del algoritmo seria la siguiente:

7	37	68	123	134	221	387	468	769

- Input:

387	468	134	123	68	221	769	37	7

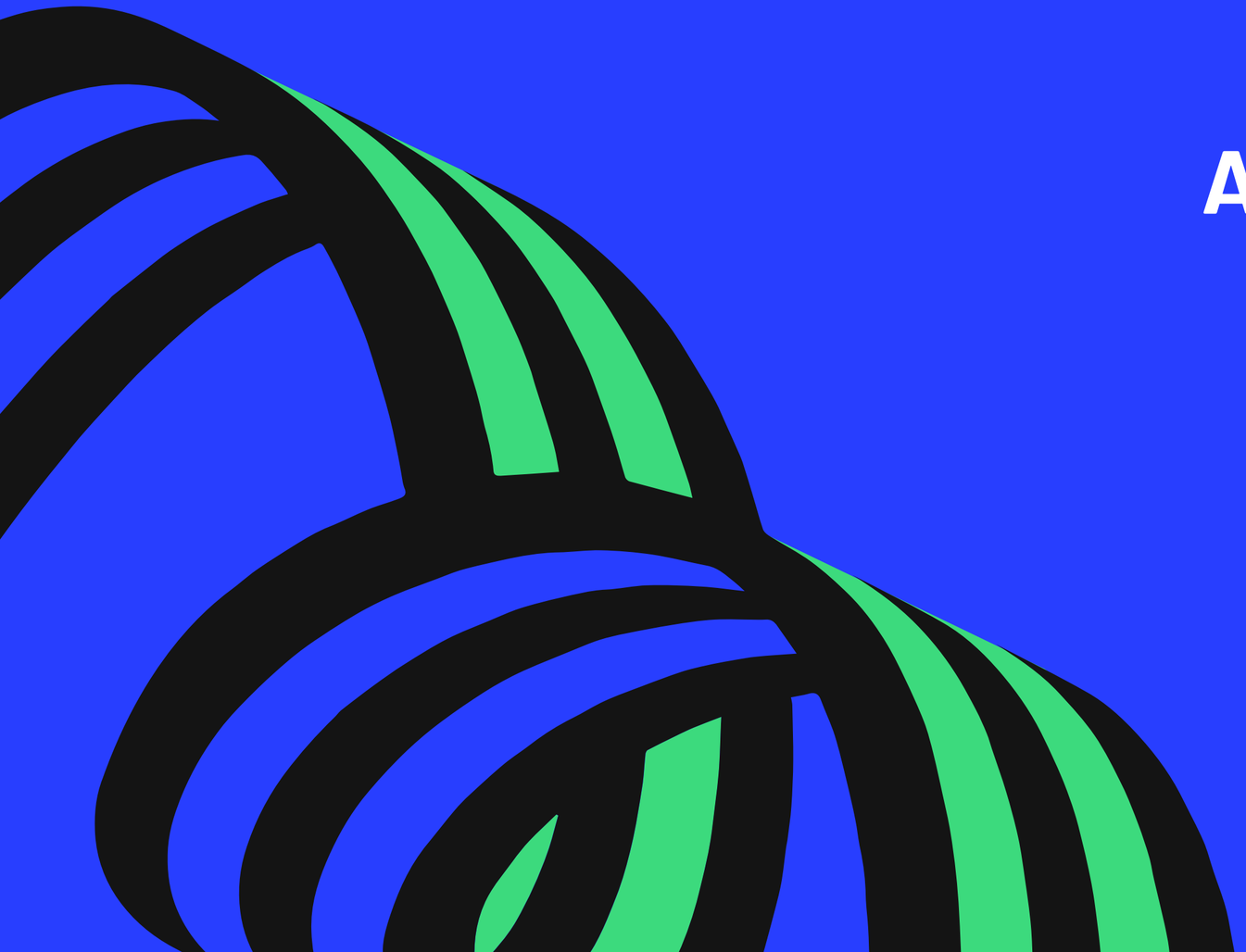
COMPARACIÓN

Primera pasada:

{9,21,4,40,10,35} --> {9,21,4,40,10,35} (no se realiza intercambio)
{9,21,4,40,10,35} --> {9,4,21,40,10,35} (intercambio entre el 21 y el 4)
{9,4,21,40,10,35} --> {9,4,21,40,10,35} (no se realiza intercambio)
{9,4,21,40,10,35} --> {9,4,21,10,40,35} (intercambio entre el 40 y el 10)
{9,4,21,10,40,35} --> {9,4,21,10,35,40} (intercambio entre el 40 y el 35)

Segunda pasada:

{9,4,21,10,35,40} --> {4,9,21,10,35,40} (intercambio entre el 9 y el 4)
{4,9,21,10,35,40} --> {4,9,21,10,35,40} (no se realiza intercambio)
{4,9,21,10,35,40} --> {4,9,10,21,35,40} (intercambio entre el 21 y el 10)
{4,9,10,21,35,40} --> {4,9,10,21,35,40} (no se realiza intercambio)
{4,9,21,10,35,40} --> {4,9,10,21,35,40} (no se realiza intercambio)



Ahora, pasemos al código:

¡GRACIAS POR LA ATENCIÓN!