QuickSort

El más popular de los algoritmos de ordenamiento

El más rápido en la mayoría de casos.

Particiona el arreglo en dos y se llama recursivamente en cada mitad.

El paso inicial es calcular el pivote. El pivote es el elemento central del arreglo.

Iterativamente compara los elementos previos y posiciones al pivote contra el valor del pivote.

Hace swap de los elementos para colocarlos en la mitad que le corresponde

Cuando los índices se cruzan, se recorre en cada mitad.

Algoritmos de ordenamiento

-Radix Sort

-Utiliza un enfoque distinto al resto de algoritmos de ordenamiento vistos hasta el momento

-Los algoritmos vistos consideran la llave (el elemento de la lista) como una unidad indivisible

->por ejemplo 100 se considera como el numero 100

-Sin embargo, RadixSort divide la llave en cada uno de sus dígitos y ordena cada uno de estos

-Radix = base / raíz del sistema numerico

- si se ordena números en base 10, radix = 10

-si se ordenan números en base 2, radix = 2

-Ordene el siguiente array

I 170 I 45 I 75 I 90 I 802 I 24 I 2 I 66 I

Arreglo intermedio con n posiciones donde n es la cantidad de elementos

0 1 2 3 4 5 6 7 8 9

I I I I I I I I I I I

0 1 2 3 4 5 6 7 8 9

I I I I I I I I I I I

Binary Search

-Busca un elemento en una lista/ array

Ordenado

-Comprara el elemento central del array contra el elemento buscado.

-Si el central == buscado, termina.

Sino:

-Central > buscado, se busca en la mitad inferior

-Central < buscado, se busca en la mitad superior