Autor
Prof. Sabrina Lizbeth Vega Maldonado

Quick Sort

El ordenamiento rápido es un algoritmo recursivo basado en la técnica de divide y venceras.

Elige un elemento de la lista al que llama pivote. Acomodar todos los elementos de la lista al lado del pivote, coloca a su izquierda todos los menores, y a su derecha otro los mayores.

Entonces el pivote ya esta en su posición (ordenado)
Ahora tenemos dos sublistas, una de menores y una de mayores. Repetir este proceso de forma recursiva para cada sublista.

Autor
Prof. Sabrina Lizbeth Vega Maldonado

Rápido (Quick Sort)

4	7	8	2	6	9	1	3	10	5

pivote

4 - 7- 8 - 2 - 6 - 9 - 1- 3 - 10 - 5
4 - 7- 8 - 2 - 6 - 9 - 1- 3 - 10 - 5
4 - 5- 3 - 2 - 1 - 9 - 6- 8 - 10 – 7
4 - 5- 3 - 2 - 1 - 6 - 9- 8 - 10 – 7

pivote

4 - 5 - 3 - 2 - 1
4 - 5 - 3 - 2 - 1
1 - 2 - 3 - 5 - 4

pivote

9 - 8 - 10 - 7
7 - 8 - 10 - 9
7 - 8 - 10 - 9

pivote

1 - 2
1 - 2

pivote

5 - 4
4 - 5

pivote

7
7

pivote

10 - 9
9 - 10

- 3 - - 6 - - 8 -

Autor
Prof. Sabrina Lizbeth Vega Maldonado

Complejidad

- **Ventajas:**
  - Rápido
  - No requiere memoria adicional.
- **Desventajas:**
  - Implementación poco complicada
  - Recursividad (utiliza muchos recursos)
  - Diferencia entre el mejor y peor caso
- **Total comparaciones**

  $C = n + n + n + ... + n = kn$

  donde $k = \log_2(n)$
- **Complejidad:**

  Mejor caso: $O(n \log_2 n)$

  Peor caso: $O(n^2)$

Autor
Prof. Sabrina Lizbeth Vega Maldonado

```
void  quickSort  ( int array[], int primero, int ultimo ){
      int pos,  i ,  j , pivote , temp ;
      i = primero;
      j = ultimo;
      pivote = array [ (primero + ultimo) / 2] ;
      do {
            while (  array [i]   <   pivote  )  {
                  i++;
            }
            while (  array [j]   >  pivote  ) {
                  j--;
             }
            if  (  i <= j  )  {
                  temp  =  array [i]
                  array  [ j ]  = array [i]
                  array  [ i ]  =   temp ;
                  i++;    j--;
            }
      }  while(i <= j );
      if(primero < j)   {
            quickSort  (array , primero, j);
      }
      if(ultimo > i)   {
            quickSort  (array , i, ultimo);
      }
}
```