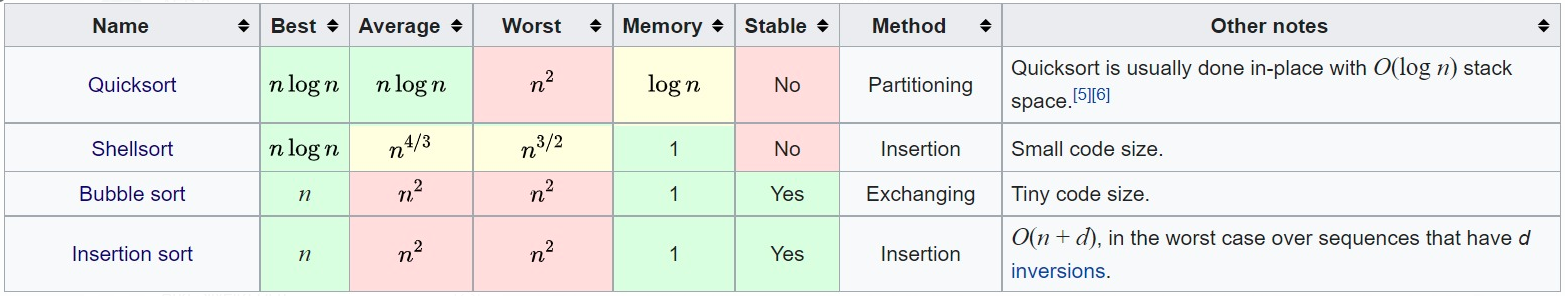
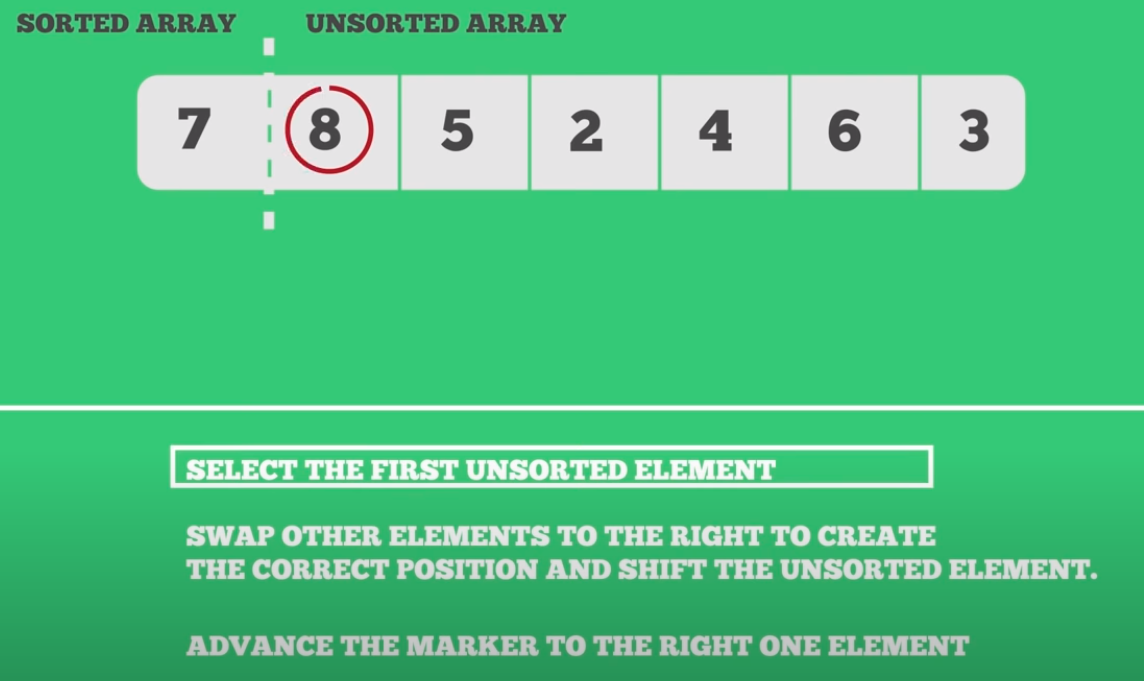
Resumen UT6



**Insertion Sort - O(n2)**



**Quick Sort - O(n2)**

Cada elemento debe saber si está en una posición de orden o no. Puede ser porque es el menor/mayor o porque todo lo de su izq es menor a el y todo lo de su der mayor.

Divide y conquistar

7 – 4 – 2 – 5 – 10 – 16 – 14 – 13 – 11

Ya se que el 10 mantiene su orden, por lo tanto divido en 2 subproblemas.

Hago lista y le pongo de ultimo dato Int MAX. (End of list marker)

Recorro desde el primero (i) pivote preguntando si es mayor que mi privote y desde End list (j) recorro preguntando si es menor, si encuentro intercambio i – j.

Cuanto j = i +1 no intercambio y encontre la posicion del pivote.

Luego se realiza de forma recursiva en cada lado.

**Shell Sort - O(n2)**

The method starts by sorting pairs of elements far apart from each other, then progressively reducing the gap between elements to be compared.

Se comparan elementos divididos por una cierta distancia

Ej N/2 = 4

Tomo lo de la derecha y comienzo a comparar arr[5] comparado arr[5-4] y veo si intercambio

Reduzco gap a N/4 y repito

Selecciono todos los elementos desde arr[1] y comparo con gap

**Bubble Sort - O(n2)**

Funciona revisando cada elemento de la lista que va a ser ordenada con el siguiente, intercambiándolos de posición si están en el orden equivocado. Es necesario revisar varias veces toda la lista hasta que no se necesiten más intercambios, lo cual significa que la lista está ordenada. Este algoritmo obtiene su nombre de la forma con la que suben por la lista los elementos durante los intercambios, como si fueran pequeñas "burbujas". También es conocido como el método del intercambio directo.