

## Ordenamientos

### Preguntas

**1. Explique cómo generó cada uno de los peores casos y por qué es el peor caso para ese algoritmo, además de mencionar el orden de la complejidad del peor caso.**

QuickSort (Peor Caso  $O(n^2)$ ): El peor caso es cuando el arreglo ya está ordenado y se escoge al primer o último elemento como pivote, porque tendría que llamarse recursivamente a la función  $n-1$  veces sorteando  $n-1$  elementos en cada llamada.

BubbleSort (Peor Caso  $O(n^2)$ ): El peor caso es cuando el arreglo está ordenado al revés del que se busca ordenar. Porque cada elemento de la lista tiene que compararse con todos los elementos de la sublista no ordenada que le siguen y cambiar posición con dichos elementos hasta llegar a su posición.

SelectionSort (Peor Caso  $O(n^2)$ ): El peor caso es el mismo que el de bubble sort. Porque cada elemento de la lista tiene que compararse con todos los elementos de la sublista no ordenada que le preceden y cambiar posición con dichos elementos hasta llegar a su posición.

MergeSort (Peor Caso  $O(n \cdot \log(n))$ ): El peor caso tiene la misma complejidad que el caso promedio. El peor caso sucede cuando los arreglos más pequeños en los que se subdivide el arreglo original están todos en desorden.

InsertionSort (Peor Caso  $O(n^2)$ ): El peor caso es el mismo que el de bubble sort. Porque cada elemento de la lista tiene que compararse con todos los elementos de la sublista no ordenada que le preceden y cambiar posición con dichos elementos hasta llegar a su posición.

**2. Explique cuáles son los mejores casos para los mismos algoritmos y cuál es su complejidad.**

Bubble Sort: Mejor caso =  $O(n)$ : Cuando el algoritmo ya está ordenado solo se recorre una vez cada elemento.

Merge Sort ( $O(n \cdot \log n)$ ): No hay una diferencia significativa entre su mejor y peor caso

Selection Sort ( $O(n^2)$ ): No hay diferencia entre su mejor y peor caso realmente en términos de tiempo.

QuickSort  $O(n \cdot \log n)$ : El mejor caso sucede cuando cada elección de pivote divide a la lista en 2 listas de tamaño igual es decir, el pivote es el elemento que va en medio de la lista ordenada.

Insertion Sort  $O(n)$ : Durante cada iteración, el primer elemento restante de la lista es comparado únicamente con el elemento de más a la derecha de la sublista ordenada.

**3. ¿En qué algoritmos la complejidad en el peor y el mejor caso es la misma? ¿Cuál es ésta?**  
Merge Sort ( $O(n \cdot \log n)$ ) y Selection Sort ( $O(n^2)$ )

**4. ¿En qué algoritmos difiere? Mencione sus complejidades en el mejor y peor caso.**

QuickSort: Mejor caso =  $O(n \cdot \log n)$ . Peor caso =  $O(n^2)$

Bubble Sort: Mejor caso =  $O(n)$ . Peor caso =  $O(n^2)$

Insertion Sort: Mejor caso =  $O(n)$ . Peor caso =  $O(n^2)$