

**Estructuras de Datos Avanzadas**

**Análisis de Eficiencia de Algoritmos de Ordenación**

**José Manuel Martínez del Campo**

**181861**

Se puede ver que tanto el BubbleSort como el SelectionSort son algoritmos muy ineficientes. De los otros tres que vimos en clase, el quickSort y el MergeSort, algoritmos recursivos, son los que mejores tiempos tienen al ordenar una lista de datos.

En esta gráfica se aprecia como no varía mucho en cuestión de tiempo losalgoritmos cuando se prueban al revés.

Al ordenar un arreglo de objetos acomodados de manera aleatoria, el quickSort presenta algunas fluctuaciones. Esto se puede causar a que, como no se sigue un orden lineal, hay ciertas acomodaciones que no van a ser tan eficientes.

El bubble sort, el selection sort tienen un tiempo muy tardado en todos los casos probados.

Tanto el bubblesort como el insertion sort tienen una eficiencia muy buena cuando el arreglo esta ordenado, por lo que solo hacen On comparaciones.

El merge y el quick sort hacen un poco más de comparaciones, pero estas no variarán tanto conforme cambie el arreglo.

El selection sort hace muchas más comparaciones.

Cuando el arreglo está invertido, el bubblesort presenta su peor eficiencia.

El Quicksort y el merge sort se mantuvieron con poca variación, ya que no dependen tanto de la posición en la que el arreglo es dado.

Aquí ya se ve como se diferencia mucho la eficiencia de unos algoritmos con otros. Por un lado tenemos el bubble, el más ineficiente. El selection y el insertion le siguen muy de cerca.

Después, tenemos el Merge y el Quick Sort, ambos fueron muy eficientes en los casos probados. Hay veces en las que uno es un poco más eficiente que el otro, dependiendo del orden en la que es dado un arreglo.