|  |  |
| --- | --- |
| Nombre | ShellSort |
| Mejor Caso | cuando el vector fue ordenado previamente |
| Complejidad en el mejor Caso | O(n log n) |
| Peor Caso | cuando los elementos están en orden inverso. |
| Complejidad en el Peor Caso | O(n^2) |
| Algoritmo *Inplace* | si |
| Algoritmo Adaptativo | no |
| Algoritmo estable | no |

|  |  |
| --- | --- |
| Nombre | MergeSort |
| Mejor Caso | Es asintóticamente optimo entonces el mejor caso y el peor son el mismo |
| Complejidad en el mejor Caso | O(n log n) |
| Peor Caso | Es asintóticamente optimo entonces el mejor caso y el peor son el mismo |
| Complejidad en el Peor Caso | O(n log n) |
| Algoritmo *Inplace* | no |
| Algoritmo Adaptativo | no |
| Algoritmo estable | si |

|  |  |
| --- | --- |
| Nombre | QuickSort |
| Mejor Caso | En el mejor caso la particion se hace en el medio del arreglo siempre por lo que las particiones forman un arbol binario completo y el tiempo del algoritmo depende del tamaño de los datos y de la altura del arbol |
| Complejidad en el mejor Caso | [O](https://es.wikipedia.org/wiki/Cota_superior_asint%C3%B3tica)(n log n) |
| Peor Caso | En el peor caso el arreglo esta semi o casi completamente ordenado entonces la particion se hace al principio del arreglo y se convierte en un insertion sort |
| Complejidad en el Peor Caso | O(n^2) |
| Algoritmo *Inplace* | si |
| Algoritmo Adaptativo | no |
| Algoritmo estable | no |