|  |  |
| --- | --- |
| Nombre del algoritmo | **ShellSort** |
| Mejor caso | Se da cuando el arreglo que se desea ordenar está casi o totalmente ordenado |
| Complejidad en el mejor caso | O(Nlog3N) |
| Peor caso | Cuando los datos del arreglo se encuentran de manera aleatoria |
| Complejidad peor caso | O(N^ 3/2) |
| Algoritmo Inplace | Algoritmo Adaptativo |
| Algoritmo Adaptativo | Si |
| Algoritmo Estable | No |

|  |  |
| --- | --- |
| Nombre del algoritmo | **MergeSort** |
| Mejor caso | Técnicamente este algoritmo se desempeña igual en todos los casos, pero el mejor caso, aunque solo sea un poco más eficiente es cuando el arreglo está totalmente ordenado |
| Complejidad en el mejor caso | O(½ N log N) |
| Peor caso | No tiene |
| Complejidad peor caso | O(N log N) |
| Algoritmo Inplace | No |
| Algoritmo Adaptativo | No |
| Algoritmo Estable | Si |

|  |  |
| --- | --- |
| Nombre del algoritmo | **QuickSort** |
| Mejor caso | Es cuando la partición siempre coge el elemento del medio como pivote |
| Complejidad en el mejor caso | O(N log N) |
| Peor caso | Cuando el arreglo ya se encuentra ordenado y además en él hay elementos repetidos |
| Complejidad peor caso | O(½ N ^2) |
| Algoritmo Inplace | Si |
| Algoritmo Adaptativo | Si |
| Algoritmo Estable | No |

|  |  |  |  |
| --- | --- | --- | --- |
|  | ShellSort (mseg) | MergeSort (mseg) | QuickSort (mseg) |
| Tiempo Ejecución1 | 0.278 | 0.121 | 0.273 |
| Tiempo Ejecución 2 | 0.167 | 0.095 | 0.298 |
| Tiempo Ejecución 3 | 0.105 | 0.103 | 0.271 |
| Tiempo Promedio (mseg): | 0.1833333333333333‬ | 0.1063333333333333‬ | 0.2806666666666667‬ |

Por el tiempo promedio de ejecución, para el caso general, el algoritmo más eficiente es MergeSort. El siguiente algoritmo en eficiencia es ShellSort. El algoritmo menos eficiente QuickSort. La diferencia entre el método de ordenamiento de ShellSort y QuickSort pudo haber sido generada por la forma en que fueron tomadas los datos, puesto fue de manera consecutiva dando a entender que el atributo aOrdenar haciendo referencia al arreglo dinámico, ya estaba ordenado, aumentando su eficiencia