Análisis algoritmos Taller 3

|  |  |
| --- | --- |
| Nombre del algoritmo | ShellSort |
| Mejor caso | Nearly sort |
| Complejidad en el mejor caso | O(n\*lg(n)) |
| Peor caso | Random |
| Complejidad en el peor caso | O(n^3/2) |
| Algoritmo *Inplace* | si |
| Algoritmo Adaptativo | si |
| Algoritmo estable | no |

|  |  |
| --- | --- |
| Nombre del algoritmo | MergeSort |
| Mejor caso | Todos tienen la misma complejidad |
| Complejidad en el mejor caso | O(n\*lg(n)) |
| Peor caso | Todos tienen la misma complejidad |
| Complejidad en el peor caso | O(n\*lg(n)) |
| Algoritmo *Inplace* | no |
| Algoritmo Adaptativo | no |
| Algoritmo estable | si |

|  |  |
| --- | --- |
| Nombre del algoritmo | QuickSort |
| Mejor caso | No hay mejor caso |
| Complejidad en el mejor caso | O(n\*lg(n)) |
| Peor caso | Todos iguales |
| Complejidad en el peor caso | O(n^2) |
| Algoritmo *Inplace* | si |
| Algoritmo Adaptativo | no |
| Algoritmo estable | no |

|  |  |  |  |
| --- | --- | --- | --- |
|  | ShellSort | MergeSort | QuickSort |
| Tiempo Ejecución 1 | 0.407 seg | 0.209 seg | 0.533 seg |
| Tiempo Ejecución 2 | 0.397 seg | 0.2 seg | 0.549 seg |
| Tiempo Ejecución 3 | 0.373 seg | 0.203 seg | 0.548 seg |
| Tiempo Promedio | 0.392 seg | 0.204 seg | 0.543 seg |

Conclusión:

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 es QuickSort.