**Información sobre algoritmos de ordenamiento**

|  |  |
| --- | --- |
| **Nombre del Algoritmo** | Mergesort |
| **Mejor caso** | Indiferente |
| **Complejidad mejor caso** | O(n\*log(n)) |
| **Peor caso** | Indiferente |
| **Complejidad peor caso** | O(n\*log(n)) |
| **Algoritmo *inplace*** | No, requiere O(n) espacio disponible |
| **Algoritmo Adaptativo** | No |
| **Algoritmo estable** | Sí |

|  |  |
| --- | --- |
| **Nombre del Algoritmo** | Shellsort |
| **Mejor caso** | Entradas parcialmente ordenadas |
| **Complejidad mejor caso** | O(n\*log(n)) |
| **Peor caso** | Entradas aleatorias |
| **Complejidad peor caso** | O(n\* log2(n)) |
| **Algoritmo *inplace*** | Sí |
| **Algoritmo Adaptativo** | Sí |
| **Algoritmo estable** | No |

|  |  |
| --- | --- |
| **Nombre del Algoritmo** | Quicksort |
| **Mejor caso** | Aleatorios |
| **Complejidad mejor caso** | O(n\*log(n)) |
| **Peor caso** | Cuando el pivote es el menor o el mayor, *few unique* |
| **Complejidad peor caso** | N2 |
| **Algoritmo *inplace*** | Sí |
| **Algoritmo Adaptativo** | No |
| **Algoritmo estable** | No |

|  |  |  |
| --- | --- | --- |
| **Tiempo en milisegundos\*** | | |
| **ShellSort** | **MergeSort** | **QuickSort** |
| 367 | 163 | 138 |
| 453 | 140 | 112 |
| 431 | 141 | 117 |
| 463 | 145 | 119 |
| 495 | 152 | 128 |
| 522 | 162 | 138 |
| **455.16** | **150.5** | **125.3** |

Por el tiempo promedio de ejecución, para el caso general, el algoritmo más eficiente es QuickSort,. El siguiente algoritmo en eficiencia es MergeSort siendo aproximadamente un 17% más lento. Finalmente, El algoritmo menos eficiente es ShellSort tardando casi 4 veces más que Quicksort y 3 veces más que MergeSort.