**Características y Análisis de**

**Algoritmos de Ordenamiento**

**Resumen de información del algoritmo ShellSort**

|  |  |
| --- | --- |
| Nombre del Algoritmo | ShellSort |
| Mejor caso | Este algoritmo presenta su mejor caso cuando los datos están casi ordenados (nearly sorted). |
| Complejidad en el mejor caso (notación O) |  |
| Peor caso | Este algoritmo presenta su peor caso cuando los datos se presentan en un orden aleatorio (random). |
| Complejidad en el peor caso (notación O) | si h=3 |
| Algoritmo *Inplace* | Sí |
| Algoritmo Adaptativo | Sí |
| Algoritmo Estable | No |

**Resumen de información del algoritmo MergeSort**

|  |  |
| --- | --- |
| Nombre del Algoritmo | MergeSort |
| Mejor caso | Este algoritmo presenta su mejor caso en todos los casos pues funciona igual en todos. |
| Complejidad en el mejor caso (notación O) |  |
| Peor caso | Este algoritmo presenta su peor caso en ninguna situación o en todas las situaciones. (funciona igual en todos). |
| Complejidad en el peor caso (notación O) |  |
| Algoritmo *Inplace* | No |
| Algoritmo Adaptativo | No |
| Algoritmo Estable | Sí |

**Resumen de información del algoritmo QuickSort**

|  |  |
| --- | --- |
| Nombre del Algoritmo | QuickSort |
| Mejor caso | Este algoritmo presenta su mejor caso en las situaciones donde los datos están casi ordenados (nearly sorted), invertidos (reversed) y en orden aleatorio (random) |
| Complejidad en el mejor caso (notación O) |  |
| Peor caso | Este algoritmo presenta su peor caso cuando los datos presentan muchos repetidos. |
| Complejidad en el peor caso (notación O) |  |
| Algoritmo *Inplace* | Sí |
| Algoritmo Adaptativo | No |
| Algoritmo Estable | No |

**Tabla de Comparación de Tiempos de los Algoritmos de Ordenamiento para el conjunto completo de Comparendos**

|  |  |  |  |
| --- | --- | --- | --- |
|  | ShellSort (mseg) | MergeSort (mseg) | QuickSort (mseg) |
| Tiempo ejecución 1 | 476 | 266 | 557 |
| Tiempo ejecución 2 | 463 | 267 | 555 |
| Tiempo ejecución 3 | 461 | 262 | 517 |
| Tiempo promedio  (mseg): | 466,6 | 265 | 543 |

**Conclusión**

Por el tiempo promedio de ejecución, para el caso general, el algoritmo más eficiente es el algoritmo de ordenamiento MergeSort. El siguiente algoritmo en eficiencia es ShellSort. El algoritmo menos eficiente es QuickSort.