CARACTERÍSTICAS Y ANÁLISIS DE ALGORITMOS DE ORDENAMIENTO

|  |  |
| --- | --- |
| Nombre del algoritmo | Shell Sort |
| Mejor caso | Es útil tanto como si los datos desordenados se encuentran cercanos o lejanos en la estructura. Su complejidad depende mucho de la secuencia de “h”s. |
| Complejidad en el mejor caso | O() |
| Peor caso | . Su complejidad depende mucho de la secuencia de “h”s. |
| Complejidad en el peor caso | O() |
| Algoritmo *InPlace* | Sí |
| Algoritmo Adaptativo | Sí |
| Algoritmo Estable | No |



|  |  |
| --- | --- |
| Nombre del algoritmo | Merge Sort |
| Mejor caso | Es más eficiente manejando medios secuenciales de acceso lento. Es la mejor opción para ordenar una lista enlazada (O(1) en espacio). |
| Complejidad en el mejor caso | O() |
| Peor caso | Es un algoritmo que no depende de la disposición inicial de los elementos en la estructura de datos, su complejidad en notación O es siempre la misma. |
| Complejidad en el peor caso | O() |
| Algoritmo *InPlace* | No |
| Algoritmo Adaptativo | No |
| Algoritmo Estable | Sí |



|  |  |
| --- | --- |
| Nombre del algoritmo | Quick Sort |
| Mejor caso | En general es bueno, depende de si el elemento escogido como pivote es un extremo inferior o superior de la estructura según el método de comparación. Para que se dé el mejor caso el elemento debe ser un elemento medio. |
| Complejidad en el mejor caso | O() |
| Peor caso | Pivote escogido es uno de los elementos extremos, inferior o superior. |
| Complejidad en el peor caso | O() |
| Algoritmo *InPlace* | Sí |
| Algoritmo Adaptativo | No |
| Algoritmo Estable | No |



|  |  |  |  |
| --- | --- | --- | --- |
|  | ShellSort (mseg) | MergeSort (mseg) | QuickSort (mseg) |
| Tiempo Ejecución 1 |  |  |  |
| Tiempo Ejecución 2 |  |  |  |
| Tiempo Ejecución 3 |  |  |  |
| Tiempo Promedio |  |  |  |



**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 .