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. El mejor caso sucede cuando… |
| Complejidad en el mejor caso | O() |
| Peor caso |  |
| Complejidad en el peor caso | O() |
| Algoritmo *InPlace* | Sí |
| Algoritmo Adaptativo | Sí |
| Algoritmo Estable | No |



|  |  |
| --- | --- |
| Nombre del algoritmo | Merge Sort |
| Mejor caso |  |
| Complejidad en el mejor caso | O() |
| Peor caso |  |
| Complejidad en el peor caso |  |
| Algoritmo *InPlace* | No |
| Algoritmo Adaptativo |  |
| Algoritmo Estable | Sí |



|  |  |
| --- | --- |
| Nombre del algoritmo | Quick Sort |
| Mejor caso |  |
| Complejidad en el mejor caso | O() |
| Peor caso |  |
| Complejidad en el peor caso |  |
| Algoritmo *InPlace* | Sí |
| Algoritmo Adaptativo |  |
| 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 \_\_\_\_. El siguiente algoritmo en eficiencia es \_\_\_\_\_\_\_ . El algoritmo menos eficiente es \_\_\_\_\_\_ .