|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nombre** | **Complejidad** | **Mejor caso** | **Complejidad** | **Peor caso** | **In** | **A** | **Es** |
| **ShellSort** | O(nlog(n)) | El mejor caso sucede cuando los datos están organizados en gran parte, dado que no tendrá que hacer swap de datos tantas veces. | O(n^2) | El peor caso sucede cuando los datos están completamente desorganizados, porque tiene que hacer muchas comparaciones y accesos en el arreglo | **X** | **X** |  |
| **MergeSort** | O(n log(n)) | La complejidad es la misma para ambos casos, dado que accede a todos valores del arreglo, o los arreglos y los compara entre sí | O(n log(n)) | La complejidad es la misma para ambos casos, dado que accede a todos valores del arreglo, o los arreglos y los compara entre sí |  |  | **X** |
| **QuickSort** | O(nlogn) | El mejor caso sucede cuando todos los elementos del arreglo están desorganizados | O(n^2) | El peor caso sucede cuando hay valores del arreglo que son únicos | **X** |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **ShellSort** | **MergeSort** | **QuickSort** |
| **Prueba 1** | 537ms | 579ms | 347ms |
| **Prueba 2** | 626ms | 313ms | 341ms |
| **Prueba 3** | 496ms | 316ms | 335ms |
| **Promedio** | 553ms | 402ms | 341ms |

## Análisis