Taller 3: Ordenamientos – Punto 8

Integrantes:

Miguel Armando Parra - 201814632

Juan Diego Gonzalez - 201911031

a) y b) Resumen de Información de los algoritmos de ordenamiento implementados

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre del algoritmo | Shell Sort | Merge Sort | Quick Sort |
| Mejor caso | Ocurre cuando el vector esta previamente ordenado. | Cuando la entrada ya está ordenada, por lo que no se realizan intercambios, pero si se necesita pasar por todos los datos. | Ocurre cuando el pivote termina en el centro de la lista, dividiéndola en dos sublistas de igual tamaño. |
| Complejidad en el mejor caso | O(n\*log2(n)) | O(n\*log2(n)) | O(n\*log2(n)) |
| Peor caso | Ocurre cuando Los elementos están en orden inverso. | Cuando el arreglo está invertido: se deben realizar muchas comparaciones, pero de igual manera se debe recorrer todo el arreglo, por lo que el orden de complejidad es casi igual. | Ocurre cuando el pivote termina en un extremo de la lista. Al hacer la partición se puede presentar esta situación. |
| Complejidad en el peor caso | O(n^(3/2)) | O(n\*log2(n)) | O(n²) |
| Algoritmo In-place | Si | No | Si |
| Algoritmo Adaptativo | Si | No | No |
| Algoritmo Estable | No | Si | No |

c)

d)

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