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) Ejercicio realizado con un archivo con los primeros 1048575 viajes (en total) del archivo completo: bogota-cadastral-2018-2-All-HourlyAggregate.csv

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hora de consulta de viajes | Número de viajes a ordenar | Shellsort(mseg) | Mergesort(mseg) | Quicksort(mseg) |
| 0 | 40952 | 36 | 21 | 21 |
| 3 | 37151 | 33 | 19 | 20 |
| 6 | 42985 | 34 | 22 | 22 |
| 9 | 41938 | 40 | 22 | 22 |
| 12 | 43885 | 38 | 23 | 22 |
| 15 | 45543 | 39 | 23 | 23 |
|  | #Viajes Promedio: 42076 | Tiempo Promedio: 36.67 | Tiempo Promedio: 21.67 mseg | Tiempo Promedio: 21.67 mseg |

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.

MergeSort y QuickSort tuvieron una eficiencia muy similar.