**Parte 3**

**Lab3**

**1.** Resultados teóricos de Complejidad Temporal en el peor caso.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Insertion sort | Shell sort | Merge sort | Quicksort |
| Complejidad O(…) en el peor de los casos para una muestra de N datos. | O(n^2) | O(n^3/2) | O(n log (n)) | O(n^2) |

**2.** Ejecución, pruebas y obtención de tiempos de ejecución de los algoritmos de ordenamiento aplicado en arreglos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tamaño de la muestra | Insertion sort  (Tiempo en mseg) | Shell sort  (Tiempo en mseg) | Merge sort  (Tiempo en mseg) | Quicksort  (Tiempo en mseg) |
| 1000 | 16 | 3 | 17 | 2 |
| 2000 | 19 | 7 | 26 | 2 |
| 4000 | 49 | 9 | 149 | 8 |
| 8000 | 76 | 11 | 190 | 11 |
| 16000 | 316 | 25 | 300 | 16 |
| 32000 | 987 | 29 | 438 | 37 |
| 64000 | 4139 | 59 | 709 | 59 |
| 128000 | 18284 | 66 | 1271 | 78 |
| 256000 | 78568 | 128 | 2625 | 118 |
| 512000 | 510508 | 351 | 4933 | 222 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Más eficiente | eficiente | No tan eficiente | Menos eficiente |
| Algoritmo | QuickSort | Shell Sort | MergeSort | Insertion Sort |

**3.** Ejecución, pruebas y obtención de tiempos de ejecución de los algoritmos de ordenamiento aplicado en listas encadenadas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tamaño de la muestra | Insertion sort  (Tiempo en mseg) | Shell sort  (Tiempo en mseg) | Merge sort  (Tiempo en mseg) | Quicksort  (Tiempo en mseg) |
| 1000 | 871 | 57 | 9 | 44 |
| 2000 | 6667 | 212 | 23 | 165 |
| 4000 | 55952 | 922 | 80 | 645 |
| 8000 | 450705 | 4073 | 302 | 2656 |
| 16000 | 446884 | 18481 | 1179 | 13791 |
| 32000 | 3640301 | 89399 | 4667 | 62708 |
| 64000 | No resultado después de 2 h. | 354118 | 19182 | 342305 |
| 128000 | x | 1615887 | 77439 | 2128378 |
| 256000 | x | Excedió | Excedió | Excedió |
| 512000 | x | x | X | x |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Más eficiente | eficiente | No tan eficiente | Menos eficiente |
| Algoritmo | MergeSort | ShellSort | QuickSort | InsertionSort |

**4.** Comparación de los algoritmos de Ordenamiento usando arreglos y listas encadenadas.

|  |  |  |
| --- | --- | --- |
|  | Estructura más eficiente | Menos eficiente |
| Insertion sort | Arreglo | Lista |
| Shell sort | Arreglo | Lista |
| Merge sort | Arreglo | Lista |
| QuickSort | Arreglo | Lista |

La estructura de datos que parece ser más eficiente teniendo en cuenta los tiempos de ejecución de los ordenamientos es: Arreglo.

Dionny S. Cárdenas

201921484

Juan Sebastián Gutiérrez

201922404