Nombres: Julian Padilla M - Pablo Pastrana

Codigos: 201913677 – 201822920

CARACTERÍSTICAS Y ANÁLISIS DE ALGORITMOS DE ORDENAMIENTO

Parte 1

* ShellSort

|  |  |
| --- | --- |
| Nombre del Algoritmo | ShellSort |
| Mejor caso | Este algoritmo presenta su mejor caso cuando los elementos estan semi ordenados. |
| Complejidad en el mejor caso (notacion o) | O(n·lg(n)) |
| Peor caso | Este algoritmo presenta su peor caso cuando los elementos se encuentran aleatoreamente. |
| Complejidad en el peor caso (notacion o) | O(n3/2) |
| Algoritmo Inplace | No |
| Algoritmo Adaptativo | Si |
| Algoritmo Estable | No |

* MergeSort

|  |  |
| --- | --- |
| Nombre del Algoritmo | MergeSort |
| Mejor caso | Este algoritmo presenta su mejor caso cuando los elementos estan aleatoreamente, semiorenados, al reves y agrupados ordenados . |
| Complejidad en el mejor caso (notacion o) | O(n·lg(n)) |
| Peor caso | Este algoritmo se caracteriza porque su peor caso es igual a su mejor caso, por ende son iguales en complejidad. |
| Complejidad en el peor caso (notacion o) | O(n·lg(n)) |
| Algoritmo Inplace | Si |
| Algoritmo Adaptativo | No |
| Algoritmo Estable | Si |

* QuickSort

|  |  |
| --- | --- |
| Nombre del Algoritmo | QuickSort |
| Mejor caso | Este algoritmo presenta su mejor caso cuando los elementos estan aleatoreamente, semiorenados y al reves. |
| Complejidad en el mejor caso (notacion o) | O(n·lg(n)) |
| Peor caso | Este algoritmo presenta su peor caso cuando los elementos se encuentran agrupados oredenados. |
| Complejidad en el peor caso (notacion o) | O(n2) |
| Algoritmo Inplace | Si |
| Algoritmo Adaptativo | No |
| Algoritmo Estable | No |

Parte 2

* Tabla de Comparación de Tiempos de los Algoritmos de Ordenamiento para el conjunto completo de Comparendos (Tiempo tomado en Waira)

|  |  |  |  |
| --- | --- | --- | --- |
|  | ShellSort(mseg) | MergeSort(mseg) | QuickSort(mseg) |
| Tiempo de Ejecución 1 | 22468 | 18480 | 70503 |
| Tiempo de Ejecución 2 | 22102 | 18572 | 72551 |
| Tiempo de Ejecución 3 | 22645 | 18600 | 77014 |
| Tiempo Promedi (mseg): | 22405 | 18550,6 | 73356 |

Parte 3

* Conclusión

Por el tiempo promedio de ejcución por el caso general, el algoritmo más eficiente es mergesort. El siguiente algoritmo en eficiencia es O(nlogn). El algoritmo menos eficiente es quicksort.