**Documentación de las características teóricas de los algoritmos de ordenamiento**

Resumen de Información de un Algoritmo de Ordenamiento

|  |  |
| --- | --- |
| **Nombre del Algoritmo** | ShellSort |
| **Mejor caso** | Este algoritmo presenta su mejor caso cuando el arreglo ya está ordenado, ya que el número de comparaciones disminuye. |
| **Complejidad en el mejor caso (notación O)** | O(N^(3/2)) |
| **Peor caso** | Este algoritmo presenta su peor caso cuando los elementos están en ordén inverso. |
| **Complejidad en el peor caso (notación O)** | O(N^(3/2))) |
| **Algoritmo *Inplace*** | Si |
| **Algoritmo Adaptativo** | Si |
| **Algoritmo Estable** | No |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del Algoritmo** | MergeSort | | |
|  | Top-Bottom | Bottom-Up | Bottom-Up Listas |
| **Mejor caso** | Este algoritmo presenta su mejor caso cuando el arreglo esta ordenado ascendente o descendentemente. | | |
| **Complejidad en el mejor caso (notación O)** | O(0.5Nlog2(N)) | | |
| **Peor caso** | Este algoritmo presenta su peor caso cuando el arreglo esta desordenado de manera que los datos se intercalen entre números muy pequeños y unos muy grandes (ej: 100, 0, 101, 1, 102, 2…). | | |
| **Complejidad en el peor caso (notación O)** | O(Nlog2(N)) | | |
| **Algoritmo *Inplace*** | No | | Si |
| **Algoritmo Adaptativo** | No | | |
| **Algoritmo Estable** | Si | | |

|  |  |  |
| --- | --- | --- |
| **Nombre del Algoritmo** | QuickSort | |
|  | Quicksort | Quicksort 3-way |
| **Mejor caso** | Este algoritmo presenta su mejor caso cuando cada partición divide el arreglo exactamente por la mitad. | Este algoritmo presenta su mejor caso cuando el número de llaves es muy pequeño. |
| **Complejidad en el mejor caso (notación O)** | O(1.39Nlog2(N)) | O(N\*num\_llaves) |
| **Peor caso** | Este algoritmo presenta su peor caso cuando las particiones se realizan separando los items más pequeños, por lo que se genera una mayor cantidad de subarreglos. | Este algoritmo presenta su peor caso cuando el número de llaves es muy grande. |
| **Complejidad en el peor caso (notación O)** | O((N^2)/2) | O(N\*num\_llaves) |
| **Algoritmo *Inplace*** | Si | |
| **Algoritmo Adaptativo** | No | |
| **Algoritmo Estable** | No | |

Comparación de los tiempos de ejecución de los tres algoritmos (6 consultas de viajes, cada una correspondiente a una hora del día, del Caso General)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hora de consulta de viajes | Número de viajes a ordenar | Shellsort (mseg) | Mergesort (mseg) | Quicksort (mseg) |
| 7 | 120034 | 167 | 30 | 15 |
| 11 | 123297 | 138 | 22 | 13 |
| 20 | 141424 | 184 | 25 | 16 |
| 15 | 130532 | 181 | 26 | 15 |
| 3 | 106036 | 112 | 18 | 12 |
| 23 | 130333 | 175 | 27 | 18 |
|  | # de viajes promedio: 125276 | Tiempo promedio:  159.5 | Tiempo promedio:  24.67 | Tiempo promedio:  14.83 |

Conclusión:

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