**Taller 3. Estructuras de Datos.**

Sergio Guillen (20191

Juan Felipe Castaño (201820865).

|  |  |
| --- | --- |
| **Nombre del Algoritmo.** | Quick Sort |
| **Mejor Caso.** | Este algoritmo presenta su mejor caso cuando al realizar el quickSort las particiones están lo más balanceadas posible. Es decir, sus tamaños son iguales o están a 1 posición uno del otro. |
| **Complejidad en el mejor caso.** | O(nLog n) |
| **Peor Caso** | Este algoritmo presenta su peor caso cuando al realizar el quickSort las particiones están muy desbalanceadas, es decir el pivote escogido para hacer la partición es el elemento más pequeño o más grande del sub arreglo. |
| **Complejidad en el peor caso.** | O(n2) |
| **Algoritmo InPlace.** | Sí |
| **Algoritmo Adaptativo.** | No |
| **Algoritmo Estable.** | No |

|  |  |
| --- | --- |
| **Nombre del Algoritmo.** | Merge Sort |
| **Mejor Caso.** | Este algoritmo presenta su mejor caso cuando el arreglo está bastante ordenado. |
| **Complejidad en el mejor caso.** | O(nLog n) |
| **Peor Caso** | Este algoritmo presenta su peor caso cuando el arreglo tiene muchos valores dispersos o desordenados en él. |
| **Complejidad en el peor caso.** | O(nLog n) |
| **Algoritmo InPlace.** | No |
| **Algoritmo Adaptativo.** | No |
| **Algoritmo Estable.** | Sí |

|  |  |
| --- | --- |
| **Nombre del Algoritmo.** | Shell Sort |
| **Mejor Caso.** | El mejor caso ocurre entre elementos separados a una distancia que va disminuyendo en cada iteración. Lo que obtenemos con esto es que cuando se emplee el algoritmo tradicional de Insert Sort, los elementos ya están ordenados relativamente, y así la cantidad de comparaciones que tiene que hacer es mucho menor |
| **Complejidad en el mejor caso.** | O(nLog n) |
| **Peor Caso** | Este algoritmo presenta su peor caso cuando los elementos están distribuidos originalmente de tal manera que la mitad mayor se encuentre en celdas pares y la mitad menor en celdas impares. |
| **Complejidad en el peor caso.** | O(n3/2) |
| **Algoritmo InPlace.** | Sí |
| **Algoritmo Adaptativo.** | Sí |
| **Algoritmo Estable.** | No |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **ShellSort(mseg)** | **MergeSort(mseg)** | **QuickSort(mseg)** |
| Tiempo de ejecución 1 | 12,304 | 3,704 | 26,589 |
| Tiempo de ejecución 2 | 12,445 | 3,647 | 26,621 |
| Tiempo de ejecución 3 | 12,455 | 3,747 | 26,512 |
| Tiempo Promedio (mseg) : | 12,401 | 3,699 | 26,575 |