Cuadro comparativo tiempos y comparaciones algoritmos de búsqueda interna Java

Las pruebas para comparar en esta tabla fueron hechas con únicamente arreglos de 100 datos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Intercambios | Comparaciones | Recorridos | Tiempos de ejecución |
| Ordenamiento Interno | | | | |
| Burbuja 0 | 2217 | 9801 | 9801 | 1.134118 |
| Burbuja 1 | 2684 | 4950 | 4950 | 0.807707 |
| Burbuja 2 | 2684 | 4872 | 4872 | 0.808987 |
| Burbuja 3 | 2684 | 4950 | 4950 | 0.805147 |
| Selección | 95 | 99 | 99 | 0.29185 |
| Inserción | 2684 | 99 | 99 | 0.216968 |
| ShellSort | 387 | 2517 | 2517 | 0.248968 |
| QuickSort |  |  |  | 0.104323 |
| Radix | 38 | 100 | 100 | 0.184966 |
| Ordenamiento Externo | | | | |
| Intercalación |  |  |  | 11.73095 |
| Mezcla directa |  |  |  | 81.059978 |
| Mezcla natural |  |  |  | 35.932076 |

Conclusiones:

Dentro del ordenamiento interno, ordenando con los mismos datos es mucho más rápido el método QuickSort.

Para el almacenamiento externo se registro mucho menos tiempo de proceso con los mismos 100 datos en intercalación porque estos estaban divididos en 2, es decir que el de mezcla directa tomaba todos estos números en conjunto y los dividía, aunque entre el de mezcla directa y el de mezcla natural, es mas eficiente el segundo ya que estos recibían el mismo parámetro