Guion 2:

* Tabla de Tiempos *MÉTODO INSERCIÓN*:

|  |  |  |  |
| --- | --- | --- | --- |
| *N* | *t ordenado* | *t inverso* | *t aleatorio* |
| 10.000 | 0,0019 | 64 | 32 |
| 20.000 | 0,00228 | 215 | 134 |
| 40.000 | 0,00670 | 232 | 116 |
| 80.000 | 0,01324 | 892 | 463 |
| 160.000 | 0,02668 | 3.342 | 2.108 |
| 320.000 | 0,05350 | 13.344 | 8.768 |
| 640.000 | 0,10755 | 63.596 | 41.110 |
| 1.280.000 | 0,21050 | 294.675 | 155.751 |
| 2.560.000 | 0,41963 | 1.179.086 | 638.034 |

Como ya se ha estudiado en clase, el método de *Inserción* presenta una complejidad lineal en el caso de que el vector ya este ordenador, sin embargo, en caso contrario, dicha complejidad se corresponde con una cuadrática.

* Tabla de Tiempos *MÉTODO SELECCIÓN*:

|  |  |  |  |
| --- | --- | --- | --- |
| *N* | *t ordenado* | *t inverso* | *t aleatorio* |
| 10.000 | 27 | 29 | 32 |
| 20.000 | 90 | 93 | 96 |
| 40.000 | 368 | 369 | 377 |
| 80.000 | 1.468 | 1.476 | 1.632 |
| 160.000 | 5.889 | 5.911 | 6.260 |
| 320.000 | 23.640 | 23.659 | 24.989 |
| 640.000 | 94.586 | 94.498 | 98.324 |
| 1.280.000 | 377.506 | 378.335 | 394.271 |
| 2.560.000 | 1.513.251 | 1.515.683 | 1.571.300 |

El método de Selección representa una complejidad O(n2) en cualquiera de los casos posibles, lo que hace que se obtengan resultados muy similares a la hora de medir los tiempos, al menos, usando este procedimiento.

* Tabla de Tiempos *MÉTODO BURBUJA*:

|  |  |  |  |
| --- | --- | --- | --- |
| *N* | *t ordenado* | *t inverso* | *t aleatorio* |
| 10.000 | 12 | 28 | 32 |
| 20.000 | 32 | 90 | 177 |
| 40.000 | 97 | 360 | 1.032 |
| 80.000 | 520 | 1.472 | 4.854 |
| 160.000 | 2.089 | 5.913 | 20.274 |
| 320.000 | 8.418 | 23.700 | 81.937 |
| 640.000 | 33.705 | 94.958 | 327.230 |
| 1.280.000 | 134.463 | 378.144 | 1.306.475 |
| 2.560.000 | 539.613 | 1.512.192 | 5.202.803 |

El ordenamiento por *Burbuja*, también presenta una complejidad cuadrática, sin embargo, debido a la mayor dificultad de ordenación que presenta el vector en orden *inverso* y *aleatorio*, hace que el tiempo de ejecución de estos se dispare, sin embargo, sigue manteniendo su complejidad O(n2).

* Tabla de Tiempos *MÉTODO QUICKSHORT MEDIANA 3*:

|  |  |  |  |
| --- | --- | --- | --- |
| *N* | *t ordenado* | *t inverso* | *t aleatorio* |
| 10.000 | 0,0771 | 0,0877 | 0,092 |
| 20.000 | 0,1611 | 0,1872 | 0,188 |
| 40.000 | 0,3359 | 0,3950 | 0,401 |
| 80.000 | 0,7171 | 0,8432 | 0,865 |
| 160.000 | 1,4751 | 1,7479 | 1,846 |
| 320.000 | 3,1235 | 3,7394 | 4,091 |
| 640.000 | 6,4649 | 7,7591 | 9,315 |
| 1.280.000 | 13,6290 | 16,3640 | 22,777 |
| 2.560.000 | 28,1202 | 33,9800 | 574,385 |

* *RapidoFatal*: Describe en el documento en que consiste este método de selección, cuándo funciona mal y cuando no y que efecto tiene en el tiempo de ejecución.

*Funciona especialmente mal en el caso de que el vector ya este ordenado, ya que coge como pivote el primer elemento del vector, lo que hace que una de las particiones este vacía y esto repercuta en el rendimiento de forma negativa (haciendo que el tiempo de ejecución sea mucho mayor)*.