Análisis Comparativo Empírico de Algoritmos de Ordenamiento

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Operaciones Realizadas | | | |
|  | Tamaño | Bubble Sort | Insertion Sort | Selection Sort | Counting Sort |
| Arreglo Aleatorio | 0 | 7 | 3 | 3 | 98309 |
| 10 | 258 | 246 | 210 | 94325 |
| 10ˆ2 | 24801 | 20496 | 15645 | 98468 |
| 10ˆ3 | 2467345 | 2004948 | 1753079 | 104204 |
| 10ˆ4 | 250306425 | 100993968 | 171534274 | 158309 |
| 10ˆ5 | 24996730560 | 19999888332 | 15783318562 | 698309 |
|  |  |  |  |  |  |
| Arreglo Ordenado | 10ˆ2 | 304 | 20496 | 18145 | 905 |
| 10ˆ4 | 30004 | 200049996 | 150064995 | 90005 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |
| Nmrs. Repetidos | 10ˆ3 | 2443605 | 1017084 | 1506495 | 104279 |
| 10ˆ5 | 24961969997 | 19997888136 | 15789392236 | 698309 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Cuando empecé a hacer el resto de los algoritmos, inicialmente puse una opción en donde se saltaba todo el ordenamiento si el largo de la lista o arreglo era igual a 0. Sin embargo quise probar sacando esa opción para ver como se comportaban los algoritmos y veo que fue lo mejor.

Puedo concluir a partir del número de operaciones y del gráfico que los órdenes vistos en clases para los distintos algoritmos estaban correctos. Los ejes del gráfico están en orden logarítmico, por lo que el **orden cuadrático** en este caso no se aprecia pues se ve como una recta con pendiente. Sin embargo, de acuerdo a los datos de la tabla, esta tasa de crecimiento si se puede apreciar. Por lo que, el comportamiento teórico comparado al empírico se confirma: Θ(N2) para BubbleSort, InsertionSort y SelectionSort.

Entre los 4 algoritmos, los tres primeros (los cuales involucran comparaciones) son los más rápidos en crecer cuando se aumenta el número de datos. Sin embargo, son más eficientes que el Counting Sort cuando se trata de arreglos de poco tamaño, ya que el Counting Sort, aunque sea más eficiente, depende completamente del número más grande del arreglo. Si este número es muy grande, el algoritmo debe recorrer un arreglo de conteo de igual tamaño. Es por eso que la línea o la tasa de crecimiento en el gráfico parece constante, pues como los datos se generan aleatoriamente, es muy probable que el número más grande sea el mayor posible (32767). Sin embargo, si uno aumentara la capacidad de la máquina y memoria, el Counting Sort podría ser incluso menos eficiente con arreglos pequeños, pero se mantendría estable con los de mayor tamaño. Comparando el comportamiento teórico vs empírico se confirma que:  Θ(N) para CountingSort.

También pude apreciar que en el caso en el que el arreglo estuviera ordenado, el algoritmo más eficiente, independiente del tamaño de los datos, es el Bubble Sort. Esto se puede apreciar en la tabla de acuerdo al número de operaciones en cada uno.