Informe TP2 – Análisis y Diseño de Algoritmos

2) Selection Sort

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Selection Sort | | | | | | | | | |
|  | Ascending | | | Descending | | | Random | | |
| Lenght | greater | swaps | Time-Ascending | greater | swaps | Time-Descending | greater | swaps | Time-Random |
| 10 | 45 | 10 | 3 | 45 | 10 | 5 | 45 | 10 | 5 |
| 50 | 1225 | 50 | 17 | 1225 | 50 | 21 | 1225 | 50 | 15 |
| 100 | 4950 | 100 | 24 | 4950 | 100 | 31 | 4950 | 100 | 22 |
| 250 | 31125 | 250 | 29 | 31125 | 250 | 62 | 31125 | 250 | 29 |
| 500 | 124750 | 500 | 73 | 124750 | 500 | 108 | 124750 | 500 | 66 |
| 1000 | 499500 | 1000 | 136 | 499500 | 1000 | 349 | 499500 | 1000 | 167 |
| 2000 | 1999000 | 2000 | 466 | 1999000 | 2000 | 1009 | 1999000 | 2000 | 480 |
| 3000 | 4498500 | 3000 | 1032 | 4498500 | 3000 | 2180 | 4498500 | 3000 | 1332 |
| 4000 | 7998000 | 4000 | 1744 | 7998000 | 4000 | 3830 | 7998000 | 4000 | 2321 |
| 5000 | 12497500 | 5000 | 2733 | 12497500 | 5000 | 5830 | 12497500 | 5000 | 3562 |

En el grafico se puede apreciar que el tiempo en este algoritmo de sorting posee un crecimiento exponencial. El peor de los casos es aquel en donde los datos se ingresan en orden descendente, se puede observar que la curva correspondiente a este caso crece mas rápido que las otras. El mejor caso es quel donde los datos se ingresan en orden ascendente, ya que en este caso no se alteran el orden de los datos.

El Selection sort usa aproximadamente N2/2 comparaciones (greater) y N cambios o swaps para ordenar una lista de longitud N. Esto se ve en la tabla

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ascending | | | Descending | | | Random | | |
| Lenght | greater | swaps | Time-Ascending | greater | swaps | Time-Descending | greater | swaps | Time-Random |
| 10 | 45 | 10 = N | 3 | 45 | 10 = N | 5 | 45 | 10 = N | 5 |

Insertion Sort

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Insertion Sort | | | | | | | | | | |
|  | Ascending | | | | Descending | | | Random | | |
| Lenght | Greater-A | | Swaps-A | time-Ascending | Greater-D | Swaps-D | time-Descending | Greater-R | Swaps-R | time-Random |
| 10 | 9 | 0 | | 3 | 45 | 45 | 4 | 31 | 27 | 5 |
| 50 | 49 | 0 | | 4 | 1225 | 1225 | 25 | 741 | 697 | 15 |
| 100 | 99 | 0 | | 5 | 4950 | 4950 | 35 | 2820 | 2739 | 27 |
| 250 | 249 | 0 | | 6 | 31125 | 31125 | 52 | 31125 | 250 | 42 |
| 500 | 499 | 0 | | 7 | 124750 | 124750 | 106 | 66740 | 66282 | 69 |
| 1000 | 999 | 0 | | 9 | 499500 | 499500 | 291 | 277733 | 276821 | 170 |
| 2000 | 1999 | 0 | | 10 | 1999000 | 1999000 | 1042 | 1114604 | 1112787 | 598 |
| 3000 | 2999 | 0 | | 12 | 4498500 | 4498500 | 2184 | 2447370 | 2444674 | 1246 |
| 4000 | 3999 | 0 | | 13 | 7998000 | 7998000 | 3802 | 4471082 | 4467468 | 2120 |
| 5000 | 4999 | 0 | | 14 | 1249750 | 12497500 | 5976 | 6725015 | 6720505 | 3075 |

En los graficos se puede ver que el tiempo crece de forma logarítmica en el caso en que los datos se ingresan en orden ascendente, y de forma exponencial en los casos en donde los datos ingresan en forma descendente o aleatoria.

El Insertion Sort usa N2/4 comparaciones aproximadamente y N2/4 cambios o swaps para el caso promedio (datos ingresados en orden aleatorio). En el peor de los casos (caso descendente), el algoritmo hace aproximadamente N2/2 comparaciones y swaps, pero por otro lado, en el mejor caso, realiza N-1 comparaciones y 0 swaps. Esto se puede ver en la tabla por ejemplo:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ascending (mejor caso) | | | | Descending(peor caso) | | | Random(caso promedio) | | |
| Lenght(N) | Greater-A | | Swaps-A | time-Ascending | Greater-D | Swaps-D | time-Descending | Greater-R | Swaps-R | time-Random |
| 10 | 9(N-1) | 0 | | 3 | 45 | 45 | 4 | 31 | 27 | 5 |

Bubble Sort

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bubble Sort | | | | | | | | | |
|  | Ascending | | | Descending | | | Random | | |
| Lenght | greater | Swaps -A | time-Ascending | greater | Swaps-D | time-Descending | greater | Swaps-R | time-Random |
| 10 | 45 | 0 | 17 | 45 | 45 | 4 | 45 | 13 | 6 |
| 50 | 1225 | 0 | 17 | 1225 | 1225 | 25 | 1225 | 617 | 16 |
| 100 | 4950 | 0 | 19 | 4950 | 4950 | 25 | 4950 | 2414 | 22 |
| 250 | 31125 | 0 | 31 | 31125 | 31125 | 62 | 31125 | 14531 | 47 |
| 500 | 124750 | 0 | 54 | 124750 | 124750 | 112 | 124750 | 54837 | 85 |
| 1000 | 499500 | 0 | 142 | 499500 | 499500 | 293 | 499500 | 219915 | 201 |
| 2000 | 1999000 | 0 | 499 | 1999000 | 1999000 | 997 | 1999000 | 888510 | 748 |
| 3000 | 4498500 | 0 | 1037 | 4498500 | 4498500 | 2189 | 4498500 | 2048240 | 1560 |
| 4000 | 7998000 | 0 | 1723 | 7998000 | 7998000 | 3804 | 7998000 | 3452614 | 2594 |
| 5000 | 12497500 | 0 | 2647 | 12497500 | 12497500 | 5909 | 12497500 | 5576024 | 4034 |

En el grafico se puede apreciar que el tiempo en este algoritmo de sorting tambien posee un crecimiento exponencial. El peor de los casos es aquel en donde los datos se ingresan en orden descendente El mejor caso es aquel donde los datos se ingresan en orden ascendente.

Se puede ver que el Bubble sort realiza aproximadamente N2/2 comparaciones en todos los casos, en el mejor de los casos hace 0 swaps, mientras que en el peor realiza aproximadamente N2/2. Por otro lado, en el caso promedio, realiza N2/2 comparaciones y N2/4 swaps. Si se observa este ejemplo:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bubble Sort | | | | | | | | | |
|  | Ascending | | | Descending | | | Random | | |
| Lenght | greater | Swaps -A | time-Ascending | greater | Swaps-D | time-Descending | greater | Swaps-R | time-Random |
| 10 | 45 | 0 | 17 | 45 | 45 | 4 | 45 | 13 | 6 |

3) En este ejercicio se utiliza el algoritmo Insertion Sort, porque es un algoritmo estable que mantiene un orden relativo, es decir, permite ordenar ordenar primero de una forma, luego de otra pero manteniendo a su vez la primera condición de ordenamiento.