Práctica 2

Algoritmia

Contenido

[Ordenación por Burbuja 2](#_Toc159759717)

[Ordenación por Selección 2](#_Toc159759718)

[Ordenación por Inserción 3](#_Toc159759719)

[Ordenación por Quicksort 3](#_Toc159759720)

[Ordenación por combinación de Quicksort e Inserción 4](#_Toc159759721)

# Ordenación por Burbuja

Midiendo tiempos del algoritmo de ordenación de burbuja obtenemos los siguientes resultados:

|  |  |  |  |
| --- | --- | --- | --- |
| n | t ordenado | t inverso | t aleatorio |
| 10000 | 611 | 1483 | 1127 |
| 20000 | 2437 | 5966 | 4507 |
| 40000 | 9725 | 23596 | 18162 |
| 80000 | 39589 | FdT | FdT |
| 160000 | FdT | FdT | FdT |

Este algoritmo posee una complejidad de O(n^2) para todos sus casos, y como podemos ver de los resultados, dicha complejidad se respeta, ya que al duplicar la cantidad de elementos, el tiempo se tiene que multiplicar por 4

# Ordenación por Selección

Los tiempos obtenidos para el algoritmo de selección son los siguientes:

|  |  |  |  |
| --- | --- | --- | --- |
| n | t ordenado | t inverso | t aleatorio |
| 10000 | 522 | 590 | 523 |
| 20000 | 2043 | 2328 | 2068 |
| 40000 | 8241 | 9254 | 8242 |
| 80000 | 32667 | 36604 | 32504 |
| 160000 | FdT | FdT | FdT |

Al igual que el anterior este posee una complejidad de O(n^2) para todos sus casos, y también cumple que los tiempos se multiplican por 4 al duplicar el número de elementos a procesar

# Ordenación por Inserción

Los tiempos resultantes han sido los siguientes:

|  |  |  |  |
| --- | --- | --- | --- |
| n | t ordenado | t inverso | t aleatorio |
| 10000 | FdF | 787 | 402 |
| 20000 | FdF | 3126 | 1572 |
| 40000 | FdF | 12544 | 6293 |
| 80000 | FdF | 49917 | 24812 |
| 160000 | FdF | FdT | FdT |
| 320000 | FdF | FdT | FdT |
| 640000 | FdF | FdT | FdT |
| 1280000 | FdF | FdT | FdT |
| 2560000 | 70 | FdT | FdT |
| 5120000 | 137 | FdT | FdT |
| 10240000 | 274 | FdT | FdT |
| 20480000 | 545 | FdT | FdT |
| 40960000 | 1102 | FdT | FdT |
| 81920000 | 2216 | FdT | FdT |

Este algoritmo posee una complejidad de O(n^2) para el caso peor y para el caso medio, en el caso mejor, que se da cuando está ordenado posee una complejidad de O(n), como podemos ver en los tiempos, el vector ordenado duplica su tiempo al duplicar el número de elementos, en los otros dos casos el tiempo se multiplica por 4 correspondiéndose con la complejidad cuadrática

# Ordenación por Quicksort

Para este algoritmo hemos obtenido los siguientes tiempos:

|  |  |  |  |
| --- | --- | --- | --- |
| n | t ordenado | t inverso | t aleatorio |
| 250000 | 119 | 148 | 172 |
| 500000 | 236 | 268 | 382 |
| 1000000 | 490 | 568 | 869 |
| 2000000 | 1019 | 1176 | 2107 |
| 4000000 | 2142 | 2404 | 6000 |
| 8000000 | 4506 | 4988 | 17060 |
| 16000000 | 9349 | 10340 | 56827 |

Este algoritmo posee una complejidad de O(n log n) para al caso mejor y el caso medio, para el caso peor, el cual es cuando el vector está ordenado y el pivote es el primer elemento se da una complejidad cuadrática, para nuestros tiempos sigue el caso casi lineal, haciendo que el tiempo se multiplique aproximadamente por 2 al duplicar los elementos a procesar

# Ordenación por combinación de Quicksort e Inserción

En esta última parte y gracias a los tiempos obtenidos, se intenta combinar los algoritmos de Quicksort e inserción, debido a que inserción es muy eficaz con vectores ya ordenados, para ello empezaremos con el algoritmo de Quicksort y cuando las particiones sea del tamaño k, se llamará al algoritmo de Inserción, gracias a esta combinación podemos tiempos inferiores:

|  |  |  |
| --- | --- | --- |
|  | Caso (k) | t aleatorio |
| Rápido | 0 | 56827 |
| Rápido + Inserción | 5 | 50275 |
|  | 10 | 44416 |
|  | 20 | 46126 |
|  | 30 | 45114 |
|  | 50 | 43020 |
|  | 100 | 30672 |
|  | 200 | 8660 |
|  | 500 | 18498 |
|  | 1000 | 35034 |

Para este análisis se ha tomado como referencia un vector de tamaño 16 millones, y generado de manera aleatoria.

Como podemos apreciar el algoritmo de Quicksort resultan un tiempo de 56827 msg, y gracias al uso de inserción podemos ver como dicho tiempo mejora, siendo un tamaño de K=200, el mejor tiempo tomado