

1 Ordenação – Noções Básicas e Algoritmos Clássicos

1.1 Lista de exercícios 1

| Algoritmo (BubbleSort) | Tempo de execução(ms) | Número de Comparações | Quantidade de Elementos |
|------------------------|-----------------------|-----------------------|-------------------------|
| | 0.105000 | 45 | 10 |
| | 0.408000 | 4950 | 100 |
| | 14.929000 | 499500 | 1000 |
| | 367.626000 | 49995000 | 10000 |

| Algoritmo (InsertionSort) | Tempo de execução(ms) | Número de Comparações | Quantidade de Elementos |
|---------------------------|-----------------------|-----------------------|-------------------------|
| | 0.104000 | 25 | 10 |
| | 0.142000 | 2501 | 100 |
| | 4.120000 | 256368 | 1000 |
| | 104.892000 | 24925183 | 10000 |

| Algoritmo (SelectionSort) | Tempo de execução(ms) | Número de Comparações | Quantidade de Elementos |
|---------------------------|-----------------------|-----------------------|-------------------------|
| | 0.178000 | 4950 | 10 |
| | 0.188000 | 4950 | 100 |
| | 6.006000 | 499500 | 1000 |
| | 155.918000 | 49995000 | 10000 |

- i) Sim o algoritmo de ordenação Insertion Sort executou 2501 comparações com um vetor de 100 elementos
- ii) Sim, foi o algoritmo que precisou de menos tempo para ser executado
- iii) Os algoritmos que fizeram mais comparações e levaram mais tempo para serem executados foram o Bubble Sort e o Selection Sort, o interessante é que nos casos onde o vetor tem tamanho 100, 1000 e 10000 ambos fizeram o mesmo número de comparações mas o tempo para o selection fazer a comparação é menor, o bubble sort demora mais pois faz uma quantidade grande de trocas, sendo o algoritmo menos eficiente entre os três.