

UTFPR - Universidade Tecnológica Federal do Paraná
Campus Campo Mourão

Aluno: Jessé Pires Barbato Rocha

RA: 2149389

Disciplina: Algoritmos e Estrutura de Dados 2

ADNP - SEMANA 1

C)

Quantidade de Comparações:

	10	100	1000	10000
SelectionSort	$0,045 \times 10^3$	$0,495 \times 10^4$	$4,995 \times 10^5$	$49,995 \times 10^6$
BubbleSort	$0,045 \times 10^3$	$0,495 \times 10^4$	$4,995 \times 10^5$	$49,995 \times 10^6$
InsertionSort	$0,026 \times 10^3$	$2,501 \times 10^3$	$254,648 \times 10^3$	$2.511,521 \times 10^4$

Tempo Gasto na Ordenação (em segundos):

	10	100	1000	10000
SelectionSort	$0,007 \times 10^{-3}$	$0,075 \times 10^{-3}$	$5,095 \times 10^{-3}$	$210,739 \times 10^{-3}$
BubbleSort	$0,006 \times 10^{-3}$	$0,198 \times 10^{-3}$	$6,346 \times 10^{-3}$	$584,086 \times 10^{-3}$
InsertionSort	$0,005 \times 10^{-3}$	$0,046 \times 10^{-3}$	$1,268 \times 10^{-3}$	$122,924 \times 10^{-3}$

d) Responda:

i) Algum algoritmo executou consideravelmente menos comparações considerando vetores com mais que 10 elementos?

R: O algoritmo de ordenação por inserção, quando a entrada era um vetor com mais de 10 elementos, destacou-se com uma quantidade de comparações consideravelmente menor.

ii) O algoritmo que executou menos comparações foi o que precisou de menos tempo para executar a ordenação?

R: Sim. O algoritmo de ordenação por inserção foi o que executou menos comparações e ordenou o vetor em menor tempo.

iii) O que tem de interessante nos dois algoritmos com maior número de comparações em relação ao tempo de execução? Explique o resultado.

R: Os dois algoritmos com mais comparações são o de ordenação por troca (ou *bubbleSort*), e o de ordenação por seleção. O interessante é que o tempo gasto por cada um é diferente. O maior tempo fica para o *bubbleSort*. Para entender a diferença de tempo, basta ter em mente como os dois funcionam. O algoritmo de ordenação por seleção encontra o maior elemento do vetor e o coloca na última posição (ou encontra o menor e o coloca na primeira posição). Ele irá repetir o processo com subvetores criados desconsiderando o(s) elemento(s) anteriormente posicionado(s) no lugar correto. Logo, tem-se certeza de que apenas uma troca será feita por varredura. Já no caso da ordenação por troca, a comparação é feita entre um elemento e seu subsequente. Logo, pode ocorrer mais de uma troca por varredura. Uma vez que realizar uma troca tem um custo, o tempo gasto para realizar a ordenação, consequentemente, aumenta.