UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Aluno: Jhonatan Guilherme de Oliveira Cunha **RA:** 2135590

Disciplina: Algoritmo e Estrutura de Dados 2

C)

NUMERO DE COMPARAÇÕES

	10	100	1000	10000
BubbleSort	$4{,}500 \times 10^{+1}$	$4{,}950 \times 10^{+3}$	$4{,}995 \times 10^{+5}$	$4{,}999 \times 10^{+7}$
InsertionSort	$2{,}400 \times 10^{+1}$	$2{,}330 \times 10^{+3}$	$2{,}466 \times 10^{+5}$	$2{,}496 \times 10^{+7}$
SelectionSort	$5{,}400 \times 10^{+1}$	$5{,}040 \times 10^{+3}$	$5{,}004 \times 10^{+5}$	$5{,}000 \times 10^{+7}$

TEMPO EM SEGUNDOS

	10	100	1000	10000
BubbleSort	0s	0s	0,17s	0,833s
InsertionSort	0s	0s	0,002s	0,214s
SelectionSort	0s	0s	0,004s	0,554s

D) Respostas:

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

R: O algoritmo InsertionSort executou um número bem menor de comparação em relação aos outros dois.

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

R: Sim, pois o mesmo não precisou ficar realizando trocas durante sua execução.

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

R: O motivo que faz o algoritmo de BubbleSort demorar mais que o SelectionSort, é que o primeiro realiza uma quantidade maior de trocas e comparações durante sua execução. Desta maneira o BubbleSort começa a se tornar ineficiente em vetores grandes.