

Trabalho 2 Programação Funcional

Parte A

Exercício 01)

Número de trocas:

Bolha Original:

Listas	l1	l2	l3	l4	l5	l6	l7	x1	x2	x3	x4	x5	x6	x7
Trocas	0	199 900 0	200 0	199 900 0	400 000 0	400 000 0	599 900 0	0	190	100	90	95	66	94

Variação 1:

Listas	l1	l2	l3	l4	l5	l6	l7	x1	x2	x3	x4	x5	x6	x7
Trocas	0	199 900 0	200 0	199 900 0	400 000 0	400 000 0	599 900 0	0	190	100	90	95	66	94

Variação 2:

Listas	l1	l2	l3	l4	l5	l6	l7	x1	x2	x3	x4	x5	x6	x7
Trocas	0	199 900 0	200 0	199 900 0	200 100 0	400 000 0	400 000 0	0	190	100	90	95	66	94

Em relação ao tempo de execução aparentemente a variação 1 é a mais efetiva, trazendo o resultado mais rápido, porém não apresentando diferença significativa no tempo para o original e a variação 2.

Exercício 02)

Número de trocas:

Seleção Original:

Listas	l1	l2	l3	l4	l5	l6	l7	x1	x2	x3	x4	x5	x6	x7
Trocas	0	?	?	?	?	?	?	0	19	10	18	10	13	19

Variação 1:

Listas	l1	l2	l3	l4	l5	l6	l7	x1	x2	x3	x4	x5	x6	x7
--------	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Trocas	0	199 9	200 0	199 9	399 9	399 9	399 9	0	19	19	18	19	18	19
--------	---	----------	----------	----------	----------	----------	----------	---	----	----	----	----	----	----

Varição 2:

Listas	l1	l2	l3	l4	l5	l6	l7	x1	x2	x3	x4	x5	x6	x7
Trocas	0	199 9	1	199 9	200 0	399 9	399 9	0	19	10	18	10	13	19

No caso do algoritmo de Seleção a Variação 2 mostrou ser um pouco mais rápido que a Variação 1, demonstrando ser um pouco mais eficiente pois a diferença não foi grande enquanto no algoritmo original apresentou uma falha no caso de listas muito grandes, onde não era possível obter nenhum resultado a não ser que a lista já estivesse ordenada.

Exercício 03)

Número de trocas:

Inserção Original:

Listas	l1	l2	l3	l4	l5	l6	l7	x1	x2	x3	x4	x5	x6	x7
Trocas	199 9	199 900 0	399 9	199 900 1	400 200 1	400 400 0	600 100 1	19	190	118	100	109	83	110

Varição 1:

Listas	l1	l2	l3	l4	l5	l6	l7	x1	x2	x3	x4	x5	x6	x7
Trocas	199 9	199 900 0	399 9	199 900 1	400 200 1	400 400 0	600 100 1	19	190	118	100	109	83	110

Não apresentou mudanças significativas em relação ao tempo do algoritmo original para a variação 1.

Exercício 04)

Número de trocas:

Quicksort Original:

Listas	l1	l2	l3	l4	l5	l6	l7	x1	x2	x3	x4	x5	x6	x7
Trocas	199 900 0	199 900 0	199 900 1	200 100 0	400 200 0	400 200 0	400 200 0	190	190	100	100	80	84	81

Variação 1:

Listas	l1	l2	l3	l4	l5	l6	l7	x1	x2	x3	x4	x5	x6	x7
Trocas	199 900 0	199 900 0	199 900 1	200 100 0	400 200 0	400 200 0	400 200 0	190	190	100	100	80	84	81

Variação 2:

Listas	l1	l2	l3	l4	l5	l6	l7	x1	x2	x3	x4	x5	x6	x7
Trocas	100 299 7	100 299 7	100 299 9	100 400 0	200 900 0	200 900 0	200 900 0	127	127	92	92	82	86	77

No caso do quicksort o algoritmo original e a variação 1 teve seu tempo de execução bem parecido apesar do original ter apresentado ser um pouco mais rápido, porém a variação 2 já foi bem mais rápido que as outras duas e assim se mostrando mais efetiva nesse ponto.

Exercício 05)

Número de trocas:

Mergesort :

Listas	l1	l2	l3	l4	l5	l6	l7	x1	x2	x3	x4	x5	x6	x7
Trocas	239 52	239 52	239 65	239 65	519 07	514 69	501 42	108	108	106	106	104	104	99

O mergesort aparentou ser mais rápido que as melhores versões da seleção e quicksort, porém a diferença é mínima no tempo de execução.