

## Análise de performance de algoritmos de ordenação - PBL 4

Este relatório compara e analisa o tempo de execução de três algoritmos de ordenação: Bubble Sort, Insertion Sort e Quicksort para 9 conjuntos de dados diferentes, sendo aleatório, crescente e decrescente, onde em cada tipo há três tamanhos, 100, 1.000 e 10.000. Os algoritmos foram implementados e testados em Java e os conjuntos de dados são números inteiros.

Desta forma, os resultados obtidos foram (em ms):

Conjunto de dados\Algoritmo		Bubble Sort	Insertion Sort	Quick Sort
Aleatório	100	1	0	18
	1.000	39	7	17
	10.000	642	165	47
Crescente	100	0	0	0
	1.000	3	2	2
	10.000	365	0	59
Decrescente	100	0	0	0
	1.000	6	3	4
	10.000	419	240	66

É possível observar que em conjuntos pequenos (100 e 1000), os algoritmos de bubble e insertion sort são mais rápidos, porém em conjuntos maiores ( $\geq 10000$ ), o quicksort é mais rápido.

Isso se dá pela complexidade dos algoritmos e uso de recursão, assim, em conjuntos pequenos a chamada recursiva do quicksort torna o algoritmo mais lento que os outros, porém conforme a complexidade dos algoritmos e o com crescimento do número de elementos, quicksort se torna mais rápido.

Algoritmo	Complexidade
Bubble Sort	$O(a \times n^2)$
Insertion Sort	$O(b \times n^2)$
Quick Sort	$O(n \times \log_2(n))$

\*Para  $a > b$

Já para os conjuntos de dados ordenados (crescente ou decrescente), em conjuntos pequenos (100 e 1000) a execução dos algoritmos é extremamente rápida pois neste caso algumas operações dos algoritmos não precisam ser realizadas, um exemplo é o insertion sort que não entra no while loop diminuindo assim a complexidade para  $O(n)$  (crescente), sendo assim o melhor caso.

Segue o print dos resultados obtidos no console:

Tipo	Tamanho	Algoritmo	Tempo (ms)
Valor inválido ignorado: Value			
aleatorio	100	Bubble Sort	1
aleatorio	100	Insertion Sort	0
aleatorio	100	Quick Sort	18
Valor inválido ignorado: Value			
aleatorio	1000	Bubble Sort	39
aleatorio	1000	Insertion Sort	7
aleatorio	1000	Quick Sort	17
Valor inválido ignorado: Value			
aleatorio	10000	Bubble Sort	642
aleatorio	10000	Insertion Sort	165
aleatorio	10000	Quick Sort	47
Valor inválido ignorado: Value			
crescente	100	Bubble Sort	0
crescente	100	Insertion Sort	0
crescente	100	Quick Sort	0
Valor inválido ignorado: Value			
crescente	1000	Bubble Sort	3
crescente	1000	Insertion Sort	2
crescente	1000	Quick Sort	2
Valor inválido ignorado: Value			
crescente	10000	Bubble Sort	365
crescente	10000	Insertion Sort	0
crescente	10000	Quick Sort	59
Valor inválido ignorado: Value			
decrescente	100	Bubble Sort	0
decrescente	100	Insertion Sort	0
decrescente	100	Quick Sort	0
Valor inválido ignorado: Value			
decrescente	1000	Bubble Sort	6
decrescente	1000	Insertion Sort	3
decrescente	1000	Quick Sort	4
Valor inválido ignorado: Value			
decrescente	10000	Bubble Sort	419
decrescente	10000	Insertion Sort	240
decrescente	10000	Quick Sort	66