

Trabalho Algoritmos de Ordenação

Grupo: Júlio César, Lorenzo Bonet

Repositório: <https://github.com/julio-czr/Sort>

Relatório de Análise de Algoritmos de Ordenação

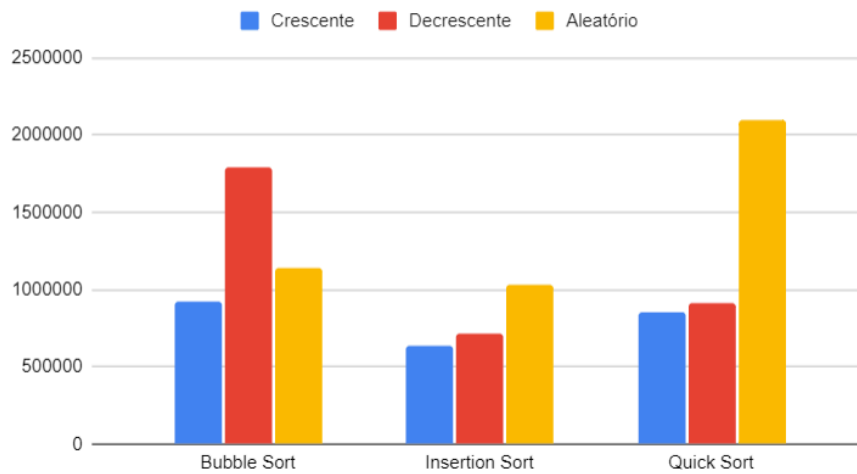
Objetivo:

O objetivo deste relatório é comparar a eficiência do algoritmo por meio do resultado do tempo de execução (nanosegundos) dos diferentes algoritmos de ordenação (Bubble Sort, Insertion Sort, Quick Sort) utilizados para ordenar diferentes tipos de conjuntos de dados.

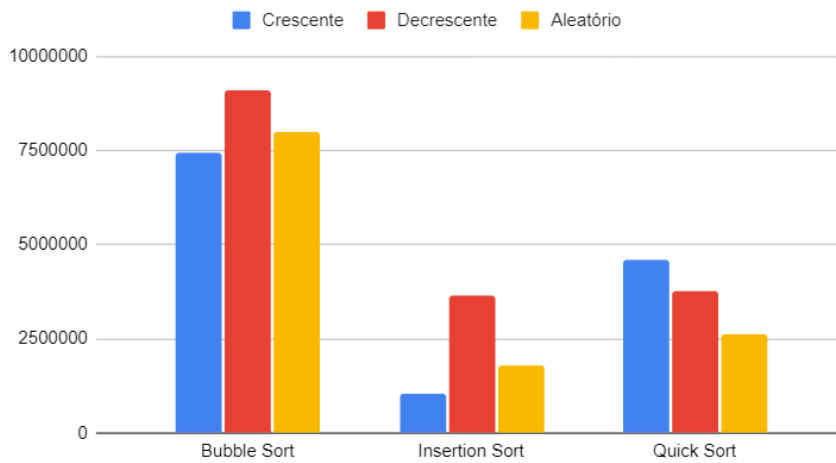
Resultados:

Conjunto pequeno de dados (100)				Legenda:
	Crescente	Decrescente	Aleatório	
Bubble Sort	924700	1792800	1140100	
Insertion Sort	638200	712400	1034500	
Quick Sort	852200	914000	2099300	
Conjunto médio de dados (1000)				
	Crescente	Decrescente	Aleatório	
Bubble Sort	7447199	9096200	7993899	
Insertion Sort	1050900	3632200	1810600	
Quick Sort	4610200	3767899	2621000	
Conjunto grande de dados (10000)				
	Crescente	Decrescente	Aleatório	
Bubble Sort	32640300	99750100	186507000	
Insertion Sort	2513100	100131600	48615901	
Quick Sort	36552000	52362700	16204400	

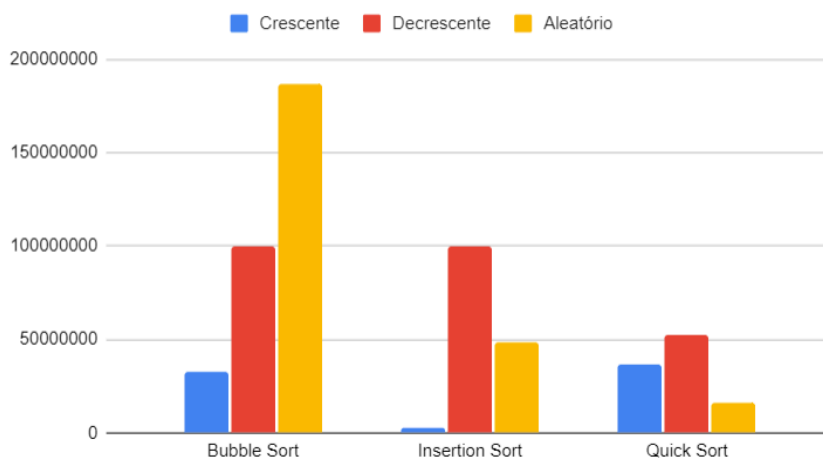
Conjunto pequeno de dados (100)



Conjunto médio de dados (1000)



Conjunto grande de dados (10000)



Conclusão:

A utilização do Insert Sort se mostra a mais recomendável para conjuntos de dados médios ou pequenos, tanto pela eficiência quanto pela facilidade de implementação, porém perde eficiência conforme o conjunto aumenta. Em conjuntos já ordenados, ele se mostra muito eficiente mesmo em conjuntos grandes, ou seja, para ordenar um novo valor em um conjunto ordenado, ele é o mais recomendado.

O Quick Sort tem melhor desempenho em conjuntos grandes, quanto maior o conjunto, maior a discrepância comparado aos outros.

Já o Bubble Sort se mostra pouco eficiente em qualquer situação.