

# ALGORITMOS DE ORDENAÇÃO

Nome do Aluno: Lucas da Rocha Silva

Rgm: 38345

Disciplina: Algoritmos e Estruturas de Dados II

Prof. Fabrício Sergio de Paula

## 1 - Introdução

Este trabalho tem por objetivo mostrar tempos de ordenação obtidos dos oito tipos de algoritmos de ordenação por tabelas e gráficos com precisão de 3 casas decimais. Mais adiante iremos ver discussões sobre os diferentes tamanhos, tempos e tipos de entradas diferentes para os algoritmos e por último a conclusão dos testes feitos. Para obter estes resultados foi usado um computador com processador i5-7300HQ de sétima geração com 8GB de memória RAM ddr4 e placa de vídeo GEFORCE GTX1050.

## 2 - Resultados obtidos

Legenda para os dados:

O - Original

M – Melhorado

PU – Pivô sendo o último elemento

PA – Pivô sendo um elemento aleatório

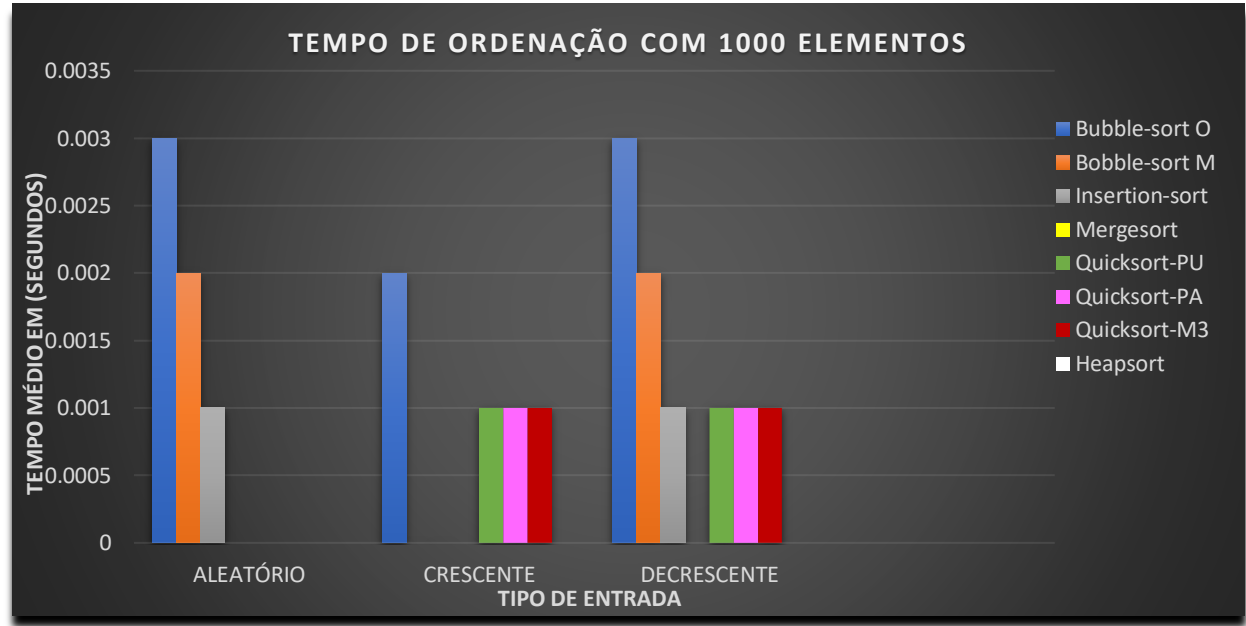
M3 – Pivô sendo mediana de três

### 2.1 - Ordenação com conjunto de 10 e 100 elementos

	TIPO DE ENTRADA		
ALGORITMOS	Aleatória	Crescente	Decrescente
Bubble-sort O	0,000	0,000	0,000
Bubble-sort M	0,000	0,000	0,000
Insetion-sort	0,000	0,000	0,000
Mergesort	0,000	0,000	0,000
Quicksort PU	0,000	0,000	0,000
Quicksort PA	0,000	0,000	0,000
Quicksort M3	0,000	0,000	0,000
Heapsort	0,000	0,000	0,000

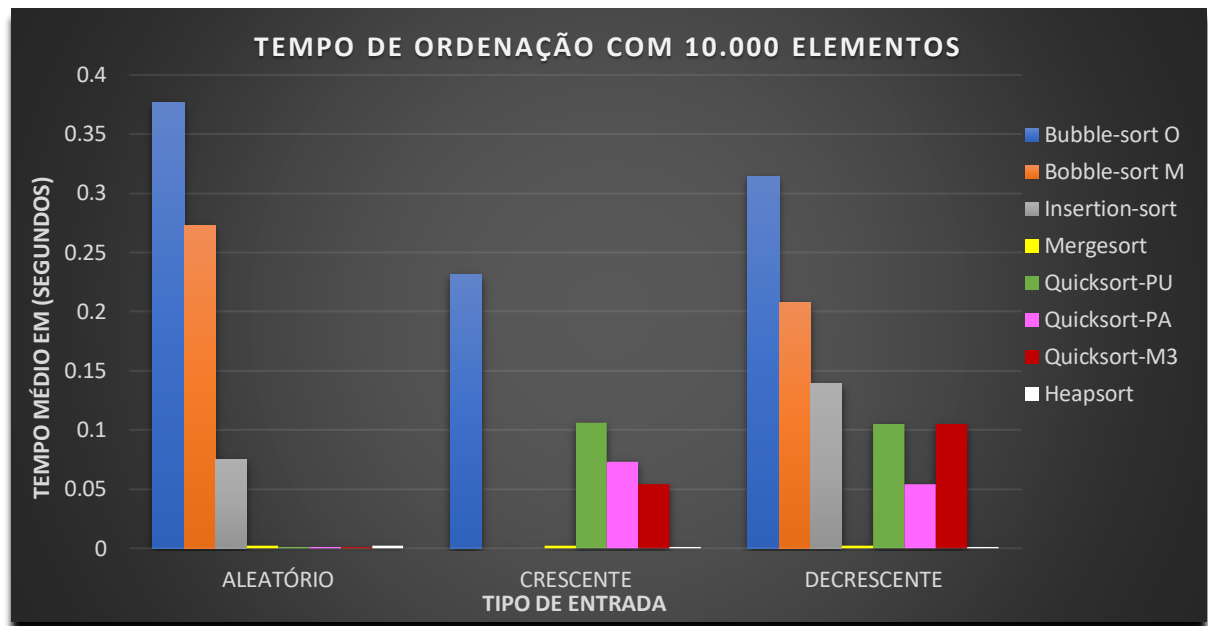
2.2 - Ordenação com 1000 elementos

	TIPO DE ENTRADA		
ALGORITMOS	Aleatória	Crescente	Decrescente
Bubble-sort O	0,003	0,002	0,003
Bubble-sort M	0,002	0, 000	0,002
Insetion-sort	0,001	0, 000	0,001
Mergesort	0,000	0,000	0,000
Quicksort PU	0,000	0,001	0,001
Quicksort PA	0,000	0,001	0,001
Quicksort M3	0,000	0,001	0,001
Heapsort	0,002	0,001	0,001



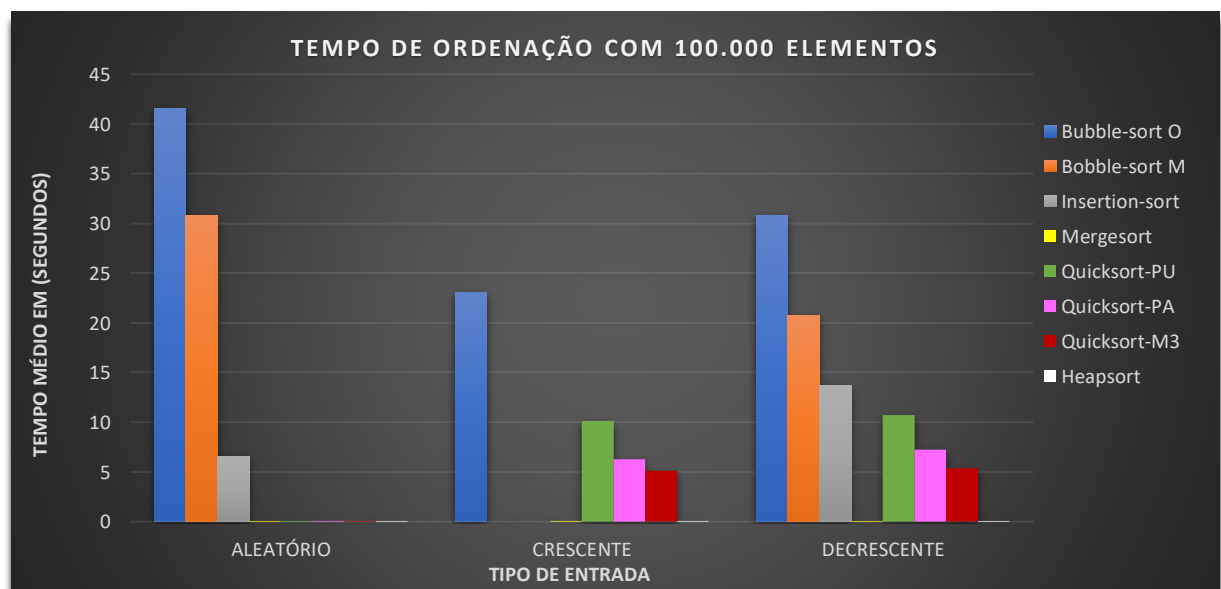
2.3 - Ordenação com 10.000 elementos

	TIPO DE ENTRADA		
ALGORITMOS	Aleatória	Crescente	Decrescente
Bubble-sort O	0, 377	0,231	0,314
Bubble-sort M	0, 273	0, 000	0,208
Insetion-sort	0,075	0, 000	0,139
Mergesort	0,002	0,002	0,002
Quicksort PU	0,001	0,106	0,105
Quicksort PA	0,001	0,073	0,054
Quicksort M3	0,001	0,054	0,105
Heapsort	0,002	0,001	0,001



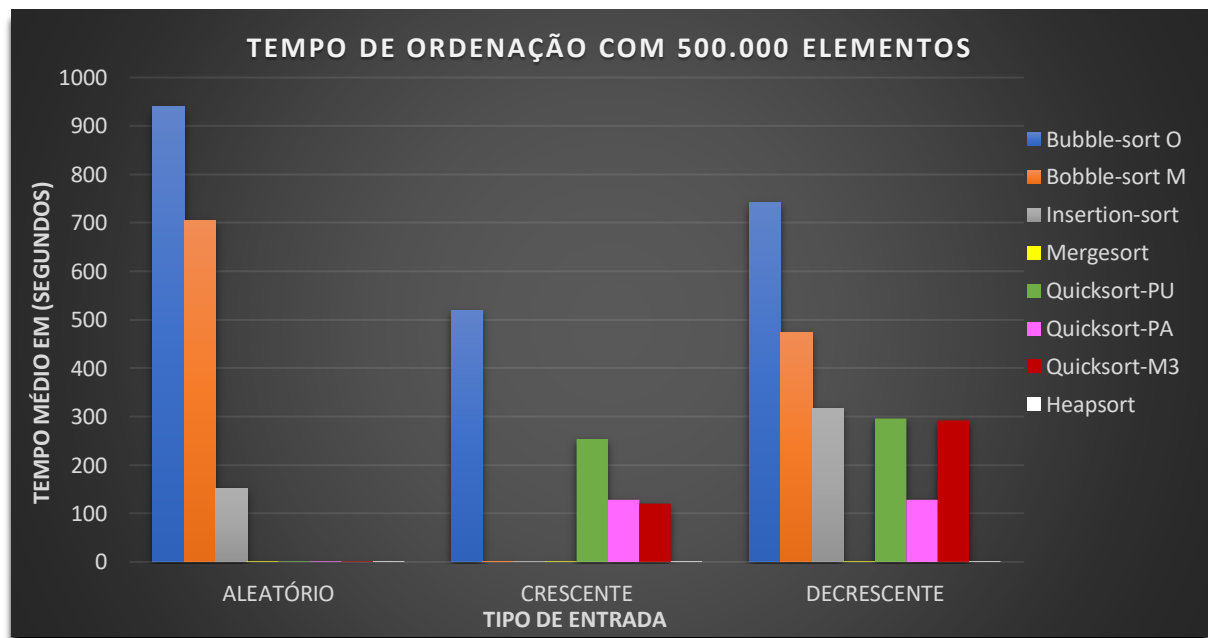
## 2.4 - Ordenação com 100.000 elementos

ALGORITMOS	TIPO DE ENTRADA		
	Aleatória	Crescente	Decrescente
Bubble-sort O	41, 561	22, 997	30, 816
Bubble-sort M	30, 802	0, 000	20, 780
Insetion-sort	6, 594	0, 000	13, 677
Mergesort	0,028	0, 018	0,018
Quicksort PU	0,015	10, 119	10, 686
Quicksort PA	0,015	6,271	7, 243
Quicksort M3	0,015	5, 136	5, 391
Heapsort	0,022	0,014	0,015



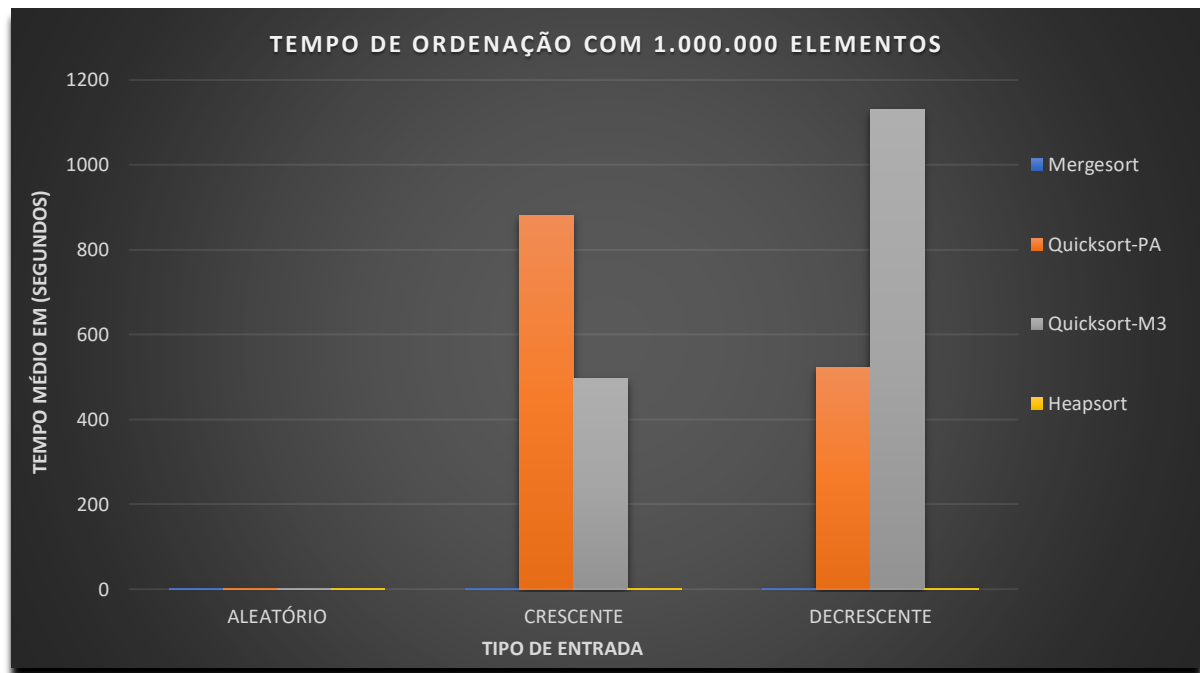
## 2.5 – Ordenação com 500.000 elementos

ALGORITMOS	TIPO DE ENTRADA		
	Aleatória	Crescente	Decrescente
Bubble-sort O	939, 187	519, 479	741, 859
Bubble-sort M	705, 039	0,001	473, 622
Insetion-sort	151, 477	0, 002	316, 283
Mergesort	0, 134	0, 090	0, 088
Quicksort PU	0, 068	253, 584	296, 318
Quicksort PA	0, 068	128, 479	128, 785
Quicksort M3	0, 076	120, 075	291, 954
Heapsort	0, 121	0, 074	0, 081



## 2.6 – Ordenação com 1.000.000 de elementos:

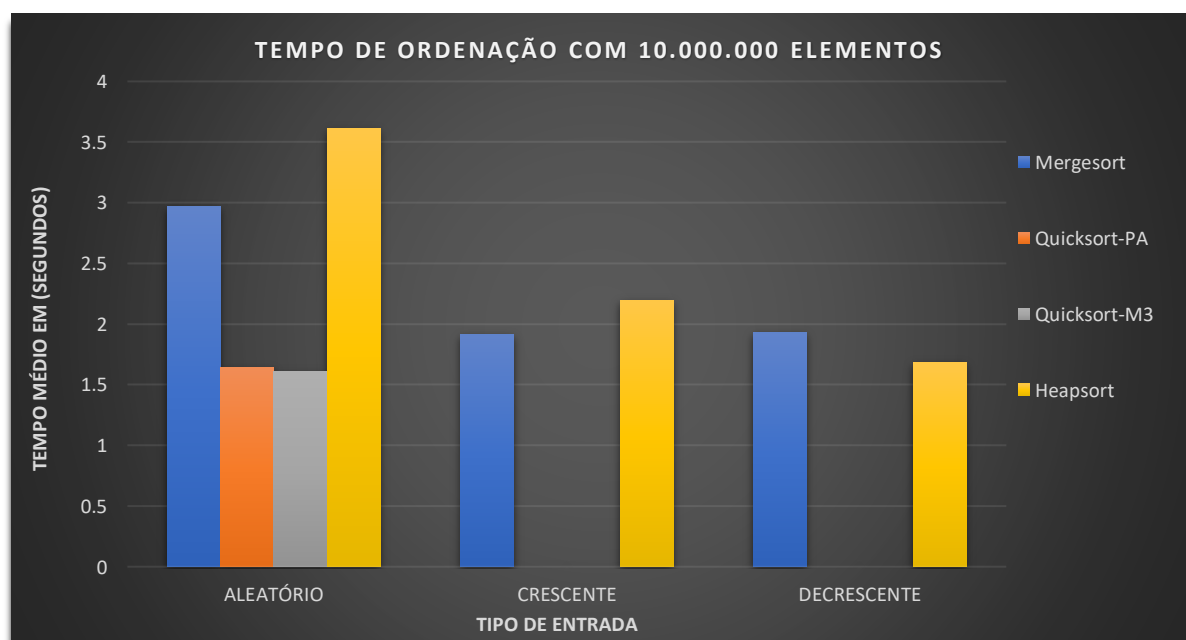
ALGORITMOS	TIPO DE ENTRADA		
	Aleatória	Crescente	Decrescente
Mergesort	0, 289	0, 175	0, 180
Quicksort PA	0, 147	881, 008	521, 817
Quicksort M3	0, 143	495, 634	1129, 444
Heapsort	0, 259	0, 155	0, 155



## 2.7 – Ordenação com 10.000.000 de elementos

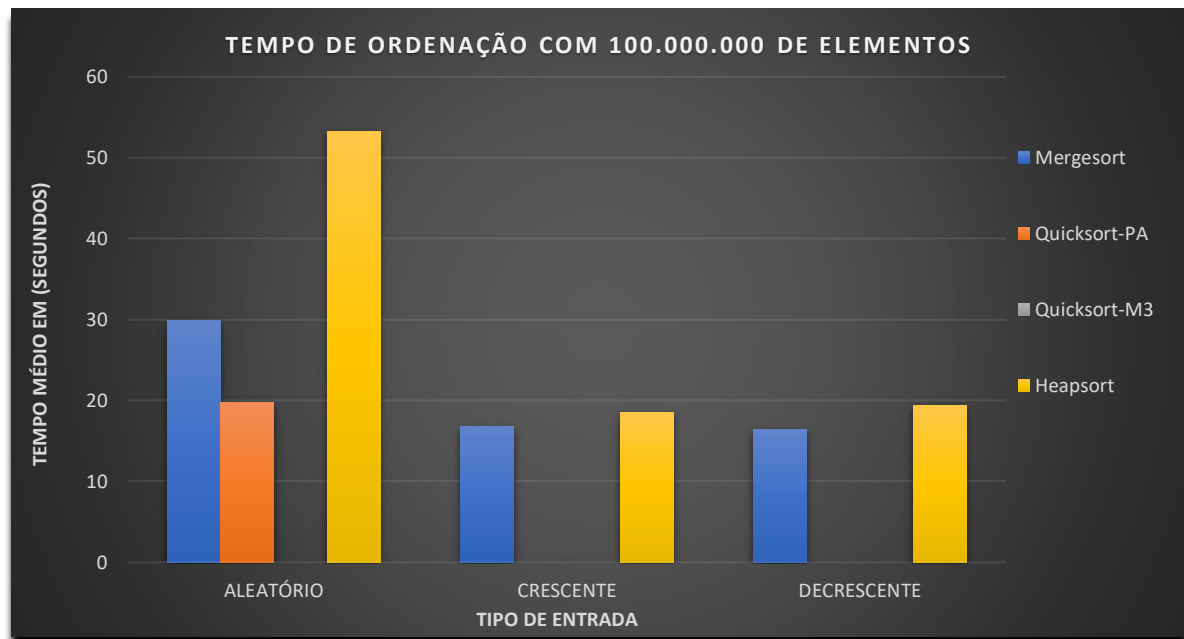
TL – Tempo muito longo para processar logo não foi possível pegar dado

ALGORITMOS	TIPO DE ENTRADA		
	Aleatória	Crescente	Decrescente
Mergesort	2, 971	1, 915	1, 932
Quicksort PA	1, 646	TL	TL
Quicksort M3	1, 607	TL	TL
Heapsort	3, 614	2, 196	1, 684



## 2.7 – Ordenação com 100.000.000 de elementos

ALGORITMOS	TIPO DE ENTRADA		
	Aleatória	Crescente	Decrescente
Mergesort	29, 914	16, 775	16, 393
Quicksort PA	19, 772	TL	TL
Quicksort M3	19, 819	TL	TL
Heapsort	53, 320	18, 558	19, 390



## 3 – Discussão

Segundo os dados obtidos podemos ver em geral que os algoritmos de ordenação mais lentos são Bubble – sort, Bubble – sort melhorado e Insertion-sort, isso porque eles tem uma complexidade de melhor caso de  $O(n^2)$  o que faz o processo de ordenação mais lento, mas com tipo de entrada crescente o Bubble-sort melhorado e o insertion-sort passa a ter sua complexidade de melhor caso para  $O(n)$ , podemos ver isso nos dados a partir de 100.000 elementos como o tempo de ordenação é bem pequeno.

Outro lado temos os algoritmos mais rápidos que são Mergesort, Quicksort e Heapsort que tem suas complexidades de melhor caso  $O(n \log n)$ , mais rápidos que os anteriores, porém os Quicksorts no tipo de entrada crescente e decrescente sua complexidade de pior caso cai para  $O(n^2)$ , podemos ver isso com detalhe a partir de 500.000 elementos, por isso que coloquei TL na tabela a partir de 10.000.000 elementos porque o tempo para ordenar é muito grande que já não conta como opção para realizar ordenações com quantidades muito grandes. Quem saiu bem nos testes foram os Mergesort e Heapsort que qualquer tipo de entrada e para qualquer tamanho de dados eles conseguiram manter suas complexidades, podemos ver o tamanho de 100.000.00.