
INSTITUTO FEDERAL GOIANO
NÚCLEO DE INFORMÁTICA
Sistemas De Informação
Estrutura de Dados II

ALGORITMOS DE ORDENAÇÃO

Giovani Barbosa dos Santos Filho

PROFESSOR: Junio Cesar de Lima

3 de outubro de 2018

1 Resultado dos Algoritmos

Utilizando os mesmos vetores de tamanho 10, 20, 30, 40, 50 em todos os algoritmos, obtemos os seguintes resultados:

Tipo de Algoritmo	10	20	30	40	50
BubbleSort:	9826,52	15608,22	23109,04	34784,58	75261,75
InsertionSort:	2078,81	3780,35	5375,10	7187,51	8496,21
ShellSort:	2375,66	5226,38	6588,30	10704,67	10369,92
QuickSort:	4338,90	6854,79	12756,48	17888,06	8818,57
MergeSort:	6482,69	11193,65	17462,03	10750,61	11485,70
HeapSort:	5942,10	14926,53	22821,00	10370,44	11631,10

Tempo em Nanossegundos

Utilizando os mesmos vetores de tamanho 60, 70, 80, 90, 100, em todos os algoritmos, obtemos os seguintes resultados:

Tipo de Algoritmo	60	70	80	90	100
BubbleSort:	37161,85	49114,96	46469,98	51610,09	70091,25
InsertionSort:	10234,75	9779,66	9362,61	10649,60	13126,42
ShellSort:	15329,19	15657,00	16249,90	18486,31	22888,64
QuickSort:	17365,28	11011,42	10578,00	11906,87	13646,53
MergeSort:	36691,31	14557,40	19044,43	15410,12	69828,19
HeapSort:	10994,42	14651,82	15302,29	16057,70	27978,94

Tempo em Nanossegundos

Utilizando os mesmos vetores de tamanho 1000, 2000, 3000, 4000, 5000 em todos os algoritmos, obtemos os seguintes resultados:

Tipo de Algoritmo	1000	2000	3000	4000	5000
BubbleSort:	418137,81	1392992,00	2652620,75	4373003,50	6794470,50
InsertionSort:	53563,81	83199,94	120131,53	150856,41	171596,58
ShellSort:	96100,12	110454,14	128081,72	142852,89	174796,36
QuickSort:	147398,69	370552,84	736617,75	1368031,25	1727190,25
MergeSort:	101351,51	157899,80	219977,98	289794,53	308359,53
HeapSort:	109002,31	180358,80	302585,78	297036,91	348612,16

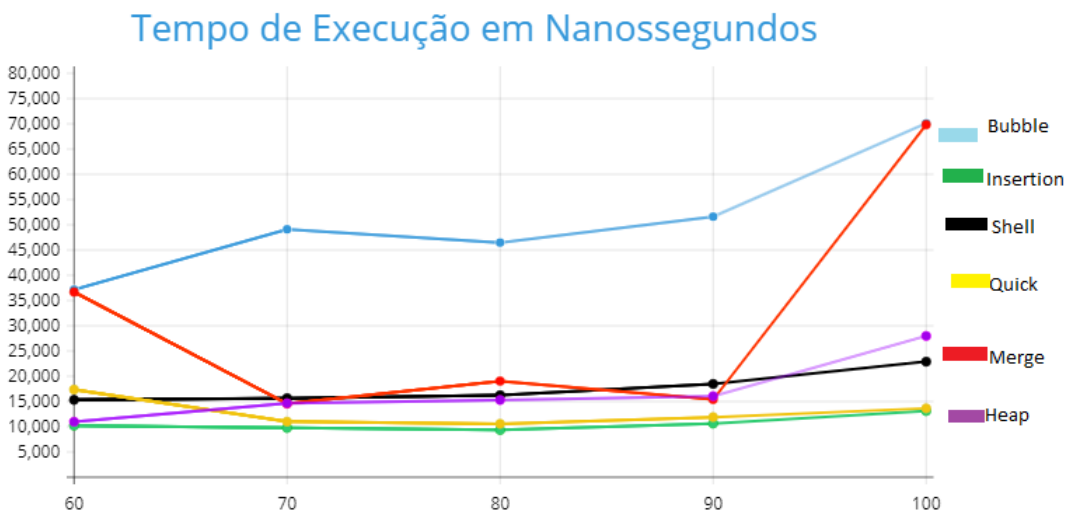
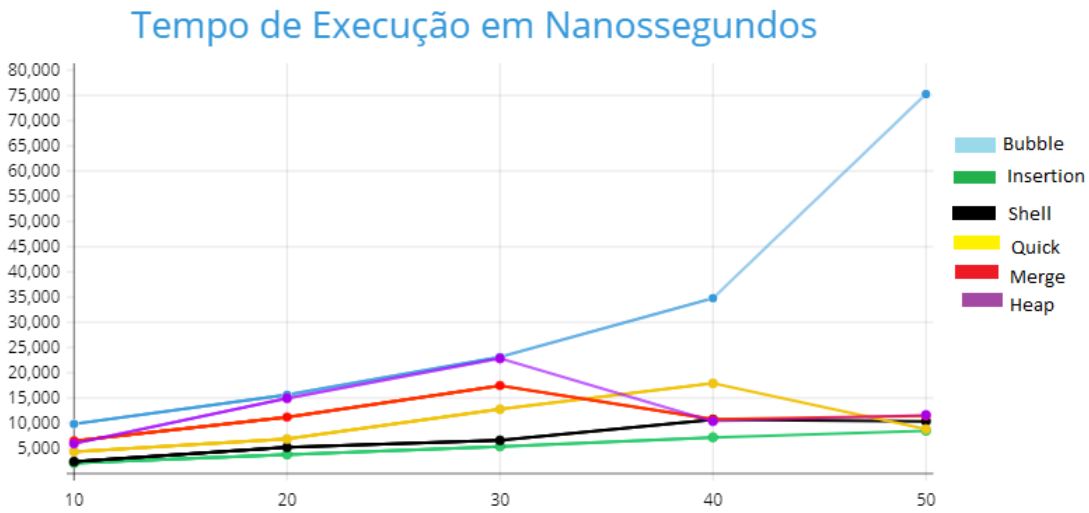
Tempo em Nanossegundos

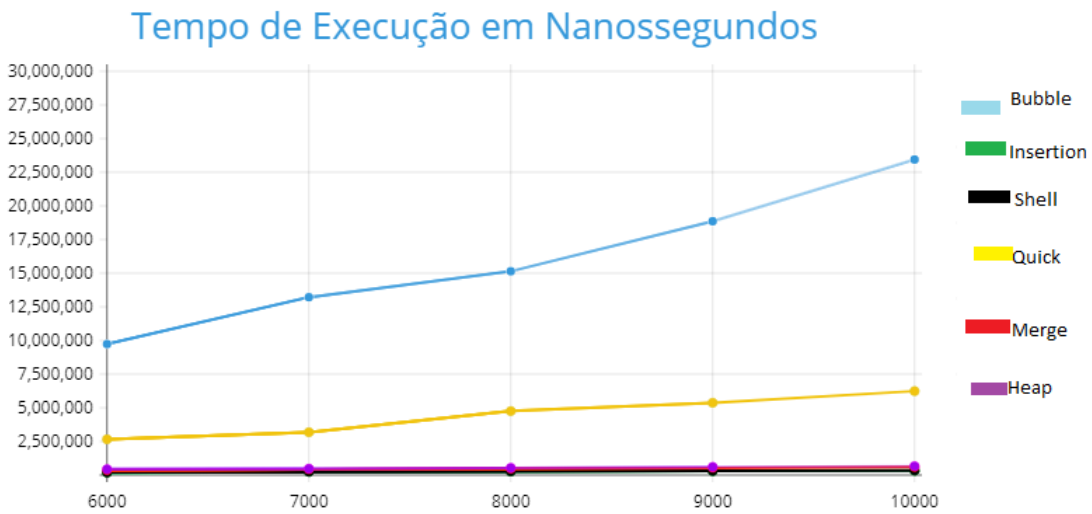
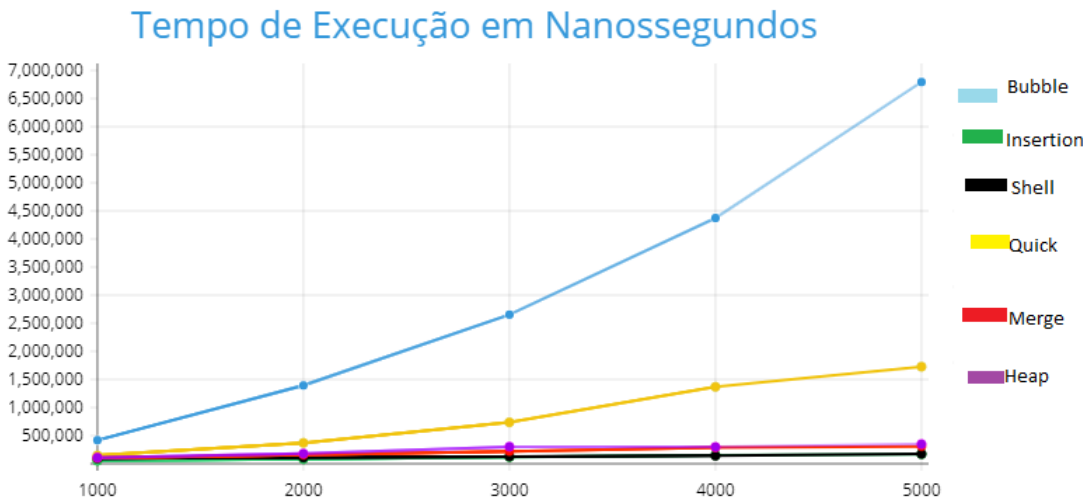
Utilizando os mesmos vetores de tamanho 6000, 7000, 8000, 9000, 10000 em todos os algoritmos, obtemos os seguintes resultados:

Tipo de Algoritmo	6000	7000	8000	9000	10000
BubbleSort:	9733176,00	13214157,00	15144973,00	18861112,00	23444644,00
InsertionSort:	190591,05	242790,50	267367,53	300153,94	330128,03
ShellSort:	195560,36	252773,36	283383,25	315359,78	349154,41
QuickSort:	2658867,75	3181924,75	4754011,00	5370520,50	6236808,50
MergeSort:	352911,03	419205,56	464939,38	531773,13	586198,38
HeapSort:	433164,72	478246,38	533417,06	602822,75	660001,75

Tempo em Nanossegundos

2 Gráficos de Tempo





3 Conclusões

Como podemos ver nos gráficos, a medida em que o número de elementos do vetor aumenta, o tempo de execução também aumenta. O algoritmo Bubble e o Selection são os mais demorados quando se trata de vetores grandes. Já o Insertion e o Shell são os melhores.

O Merge, Heap e o Quick mantém uma taxa de tempo parecida em quase todos os tamanhos de vetores.

Conclui-se que o melhor algoritmo de ordenação para esses casos é o HeapSorte, pois independente do aumento da quantidade de números no vetor, ele não deixa de ser eficiente.

4 Informações do Computador

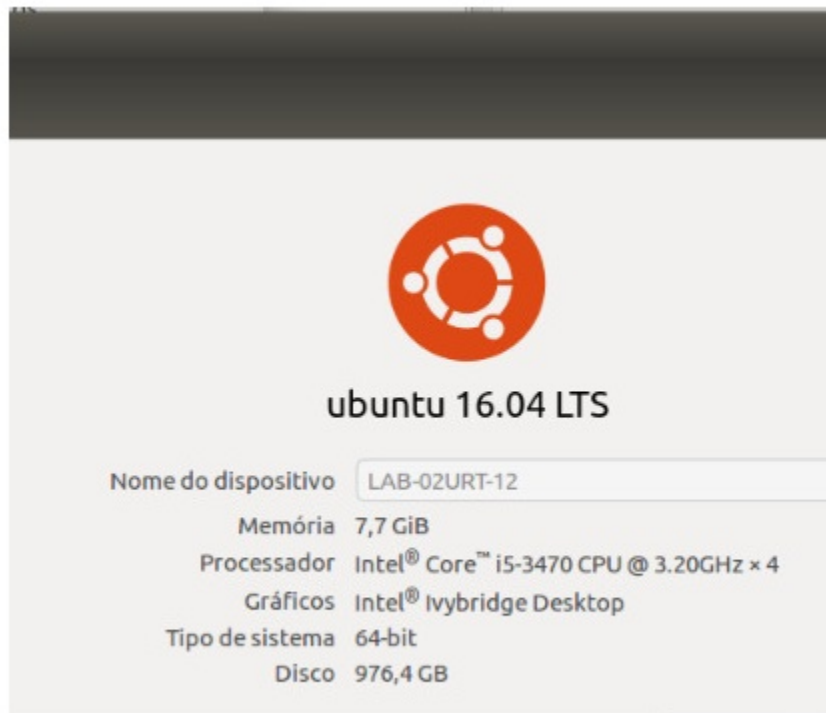


Figura 1: Informações do computador