

Relatório Lab3

David Costa Pereira

15 de setembro de 2022

1 Considerações

Para obter os dados deste relatório foi usado o gerador de strings fornecido no classroom.

2 Bubble Sort

Avaliando o tempo de execução e o número de comparações feitas pelo algoritmo de ordenação bubble sort para os dados da Tabela 1, observa-se que o tamanho máximo que o algoritmo consegue executar em dois segundos é aproximadamente 11000.

Tamanho da entrada	Tempo (s)	Comparações
2000	0.070	3998000
3000	0.136	8997000
4000	0.246	15996000
5000	0.400	24995000
6000	0.598	35994000
7000	0.792	48993000
8000	1.087	63992000
9000	1.367	80991000
10000	1.650	99990000
11000	2.003	120989000
12000	2.365	143988000
13000	2.742	168987000
14000	3.162	195986000
15000	3.710	120989000
16000	4.162	120989000
17000	4.577	120989000
18000	5.380	120989000
19000	5.823	120989000
20000	6.665	120989000
21000	7.277	120989000

Tabela 1: Entradas e seus respectivos tempo de execução e o número de comparações feitas.

Nota-se que o tempo de execução, Figura 1, segue a mesma tendência do número de comparações, Figura 2, feitas para cada entrada. Fato este que fortalece a proposta de que as comparações feitas são os fatores mais relevantes para o tempo de execução do programa.

Outro fato importante de se avaliar é o comportamento limitante do gráfico de tempo por tamanho da

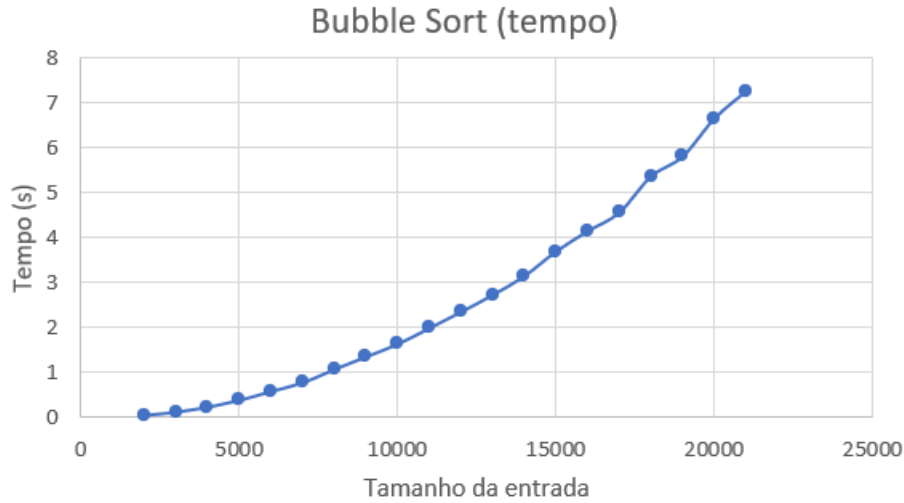


Figura 1: Gráfico do tempo de execução por tamanho da entrada.

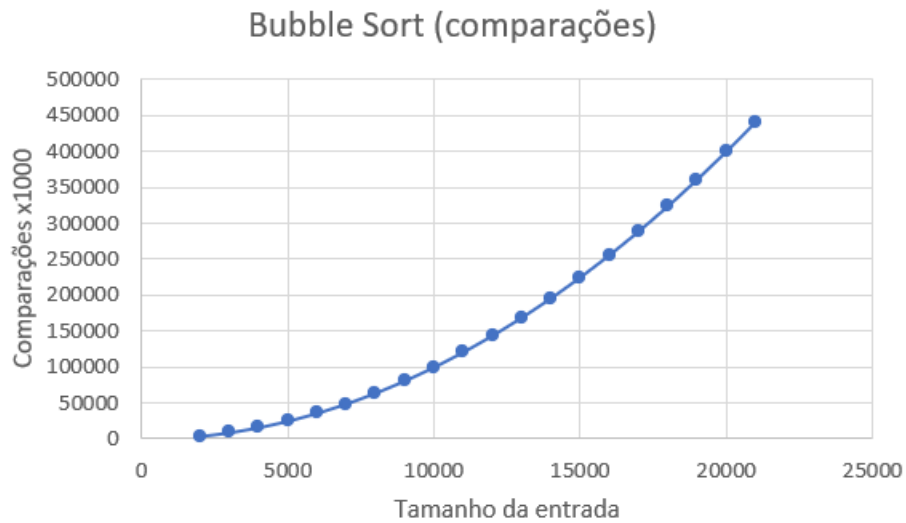


Figura 2: Gráfico do número de comparações por tamanho da entrada.

entrada. Verifica-se, veja a Figura 1 ou Figura 2, que a curva apresenta comportamento quadrático, ou seja $O(n^2)$, o que confirma o previsto pela teoria.

3 Merge Sort

Feito os testes com os dados da Tabela 2, observou-se que o algoritmo merge sort conseguiu executar, em 2 s, uma entrada de tamanho aproximadamente 3400000.

Analisando os graficos de tempo e numero de comparacoes, Figura 3 e Figura 6, verifica-se que a relação entre o tempo gasto e o número de comparações estão coerentes, ou seja, o tempo gasto está

Tamanho da entrada	Tempo (s)	Comparações
200000	0.112	3272562
400000	0.223	6946073
600000	0.328	10768514
800000	0.438	14690147
1000000	0.554	18673518
1200000	0.673	22737772
1400000	0.835	26845999
1600000	0.941	30979976
1800000	1.055	35157235
2000000	1.172	39347647
2200000	1.292	43588489
2400000	1.463	47874250
2600000	1.561	52172832
2800000	1.667	56490476
3000000	1.801	60819988
3200000	1.906	65164040
3400000	2.016	69523643
3600000	2.160	73909616
3800000	2.323	78302282
4000000	2.242	82692349
100000000	75.148	2532831405

Tabela 2: Entradas e seus respectivos tempo de execução e o número de comparações feitas.

proporcional ao número de comparações.

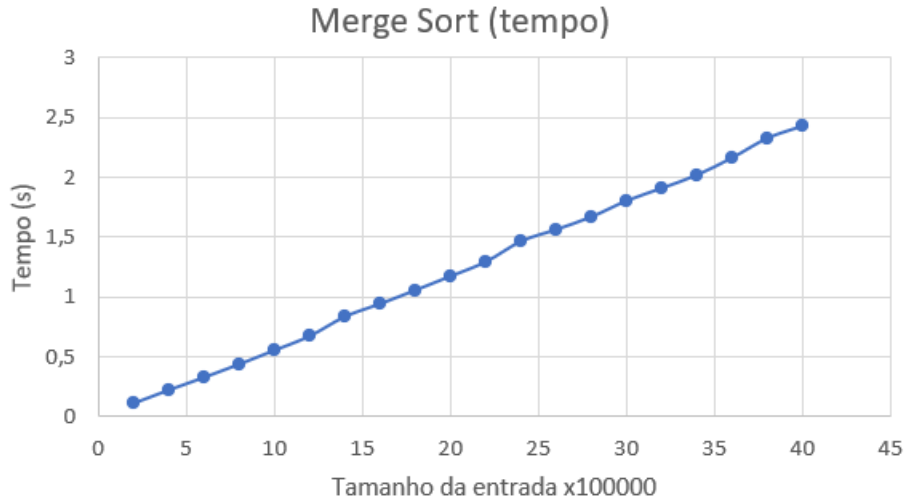


Figura 3: Gráfico do tempo de execução por tamanho da entrada.

Pode se pensar, vendo os gráficos de tempo e número de comparações, que o merge sort é da ordem de n , isso ocorre pelo fato de que a função $f(x) = x \log(x)$ tem a taxa de variação da inclinação da reta tangente muito baixa para valores de n muito grandes, em outras palavras $\frac{d^2 f}{dx^2} = \frac{1}{\ln(2)x} \simeq 0$ para $x \gg 1$. Pode-se observar que para uma entrada de tamanho 100000000, temos o tempo de execução

valendo 75.148 s, e para 1000000, temos 0,554 s, supondo que o algoritmo seja $O(n)$, deveríamos ter $0.554 \times 100 = 55.400s$ para a maior entrada, mas na realidade temos 19.748 s a mais. Para $O(n \log(n))$, temos uma boa concordância entre os dados, salvo pequenos desvios.

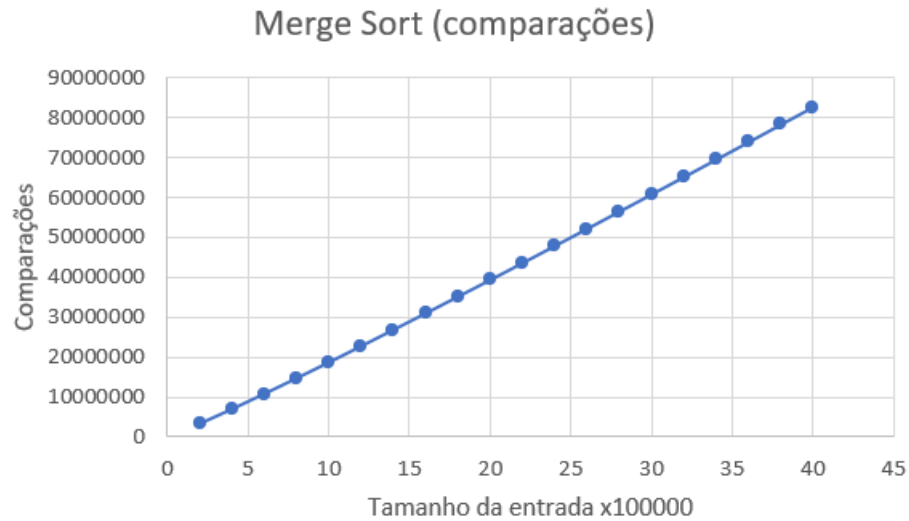


Figura 4: Gráfico do número de comparações por tamanho da entrada.

4 Quick Sort

Feito vários testes, obteve-se a Tabela 3 para o algoritmo quick sort, o algoritmo é capaz de processar uma entrada de tamanho 3500000 em aproximadamente 2 segundos.

Tamanho da entrada	Tempo (s)	Comparações
1000000	0.518	33639537
2000000	1.131	80027364
3000000	1.675	128159226
3500000	1.978	152526550
4000000	2.327	185264799
5000000	3.13	249949904
7500000	5.217	445828161
10000000	7.753	702303239
12500000	10.703	989684270
15000000	14.208	1339517196
17500000	18.106	1711545026
20000000	22.135	2133799644

Tabela 3: Entradas e seus respectivos tempo de execução e o número de comparações feitas.

Observando os gráficos, Figura e Figura, verifica-se que a relação entre o tempo de execução e o número de comparações feitas está coerente, além disso pode-se notar que o algoritmo é $O(n^2)$, o que condiz com os resultados teóricos.

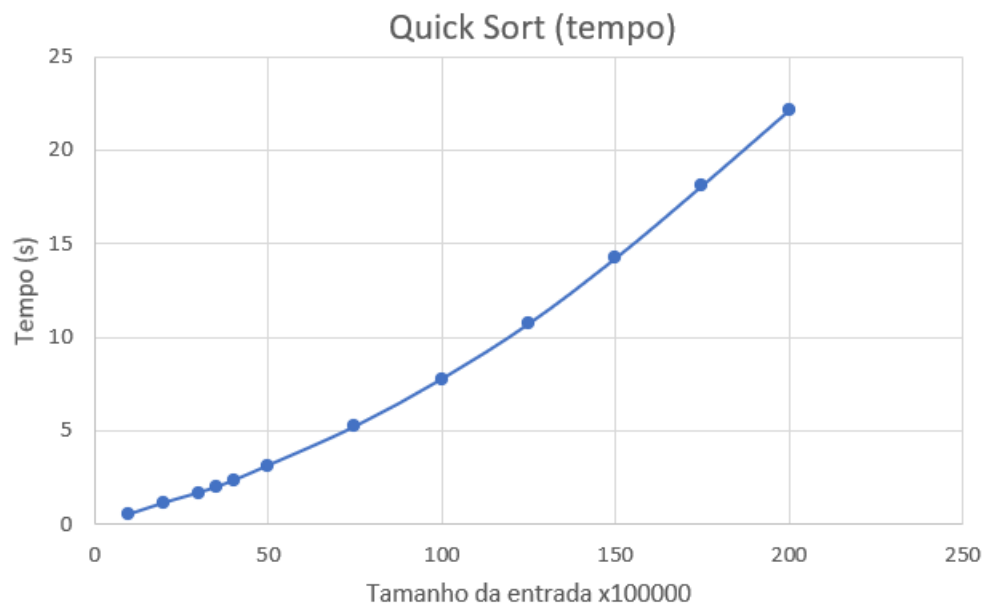


Figura 5: Gráfico do tempo de execução por tamanho da entrada.

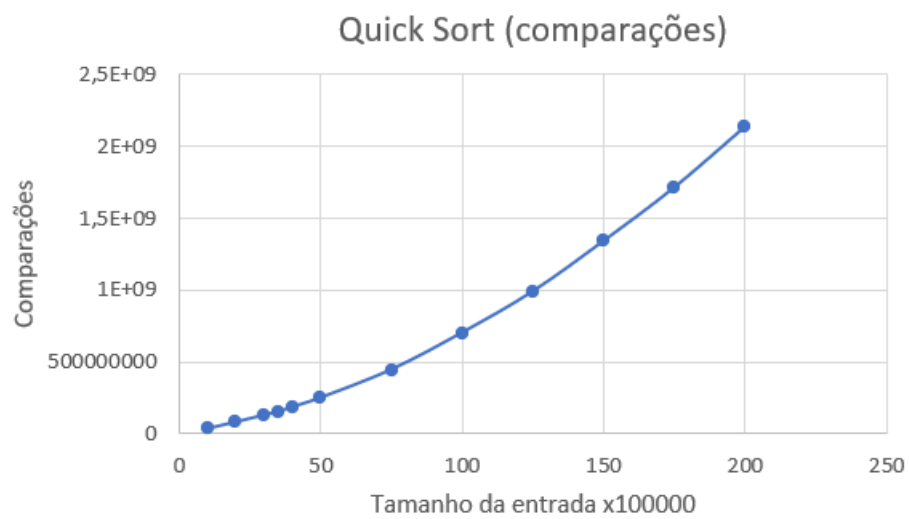


Figura 6: Gráfico do número de comparações por tamanho da entrada.