

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

PROFESSOR: MARCELO ZAMITH

Trabalho de

Arquitetura de Computadores I

Tabelas, gráficos e interpretações dos resultados dos algoritmos

ALUNA: JULIA RODRIGUES DA COSTA

Considerações sobre o ambiente

Os resultados dos dois primeiros algoritmos que serão apresentados foram obtidos através de um desktop com processador Intel Core 3 e 4Gb de RAM.

Os resultados dos dois últimos foram obtidos através de um notebook com processador Pentium Dual Core com 2Gb de RAM.

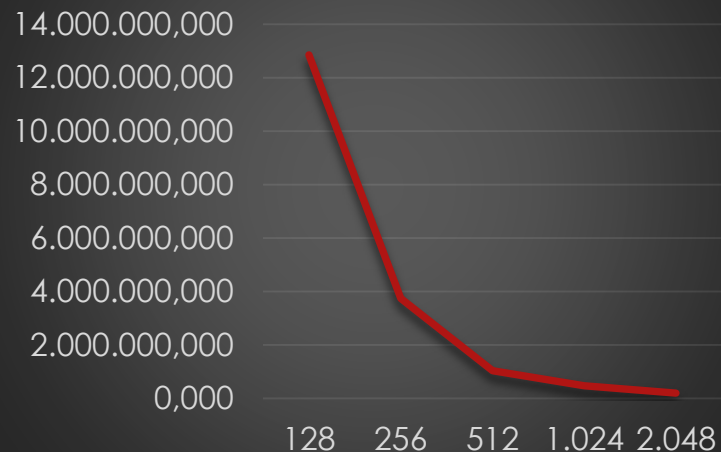
Multiplicação de Matrizes

Ordem da Matriz	Memória	Tráfego de Dados	Ciclos
128	131.072	12.855.069,242	79.348.379,700
256	524.288	3.739.238,626	1.103.728.723,100
512	2.097.152	1.037.859,471	15.553.067.632,900
1.024	8.388.608	473.085,038	137.877.015.550,100
2.048	33.554.432	195.516,325	1.328.191.240.095,500

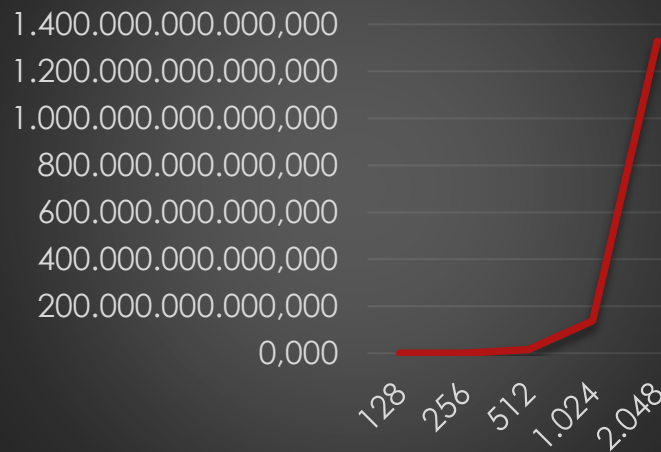
Flips	Flops	Tempo de Execução aprox. (em segundos)
6.333.823,100	6.317.734,600	0,031
50.294.123,900	50.407.387,200	0,421
403.678.006,200	412.258.861,400	6,062
3.231.114.213,900	3.321.243.921,900	53,195
26.749.901.580,400	25.767.007.016,600	514,859

Multiplicação de Matrizes

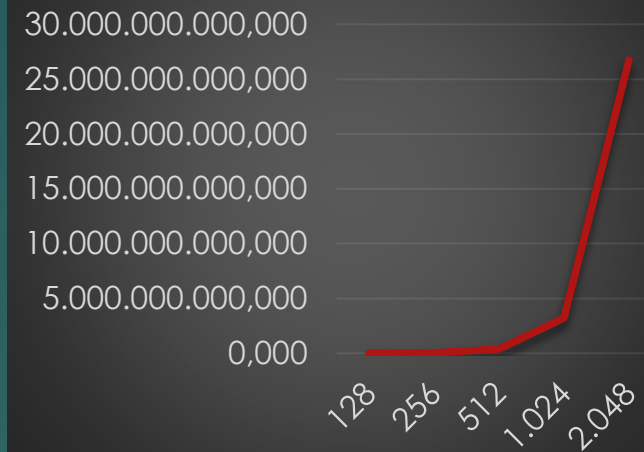
Tráfego de Dados



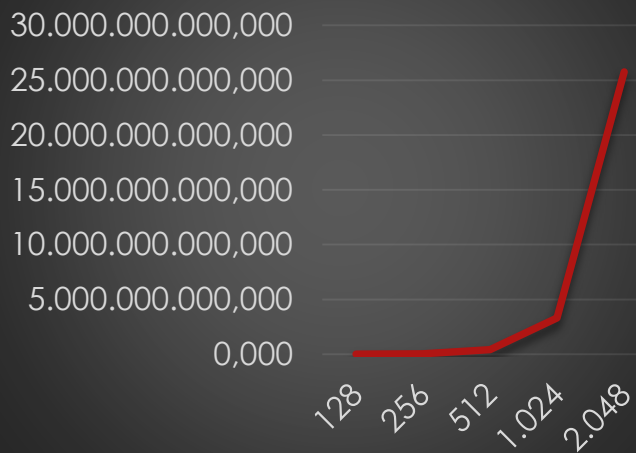
Ciclos



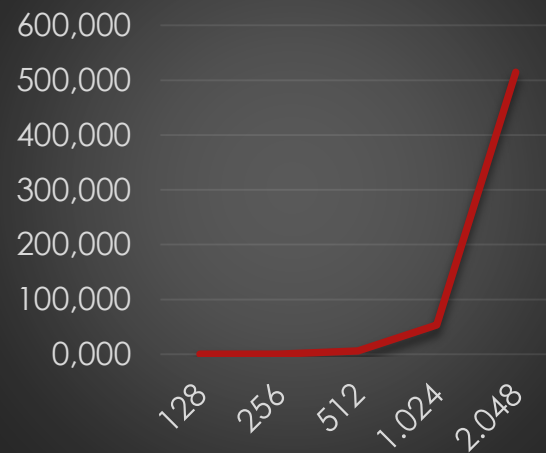
Flips



Flops



Tempo



Multiplicação de Matrizes

Entre os dois algoritmos que envolvem cálculo matricial, este é o mais lento. Além disso, é o mais lento dos quatro algoritmos que serão apresentados. Sua complexidade é $O(n^3)$.

Percebe-se pelo gráfico do tempo um aumento significativo de valores a partir de ordens após 512. O aumento exponencial ocorre um pouco após a matriz ultrapassar a ordem 1024, tornando a execução do algoritmo mais lenta.

Os gráficos do total de ciclos de clock gastos, dos flips e dos flops seguem os padrões de aumento do gráfico do tempo, ou seja, têm os picos verticais quando a matriz atinge as mesmas ordens.

Podemos perceber também que o tráfego de dados, antes da ordem 256, diminuía rapidamente. Porém, após este valor, o tráfego de dados passa a diminuir de forma cada vez mais lenta, tendendo a uma maior estabilidade quando a matriz atinge a ordem 1024.

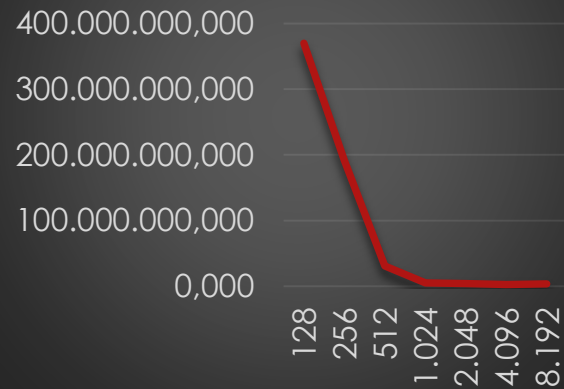
Fatoração LU

Ordem da Matriz	Memória	Tráfego de Dados	Ciclos
128	396.288	369.982.261,227	2.788.550,100
256	1.579.008	192.320.378,062	21.096.930,900
512	6.303.744	30.832.404,248	519.966.000,600
1.024	25.190.400	5.247.999,985	12.403.040.350,980
2.048	100.712.448	4.330.027,211	74.461.004.719,998
4.096	402.751.488	2.756.474,329	459.110.000.001,000
8.192	1.610.809.344	3.831.638,457	1.803.550.000.001,000

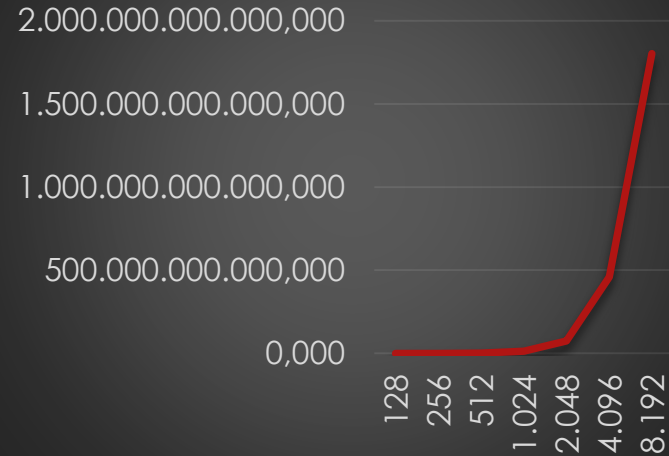
Flips	Flops	Tempo de Execução aprox. (em segundos)
1.478.569,090	1.468.139,300	0,001
11.480.803,000	11.480.501,100	0,008
90.508.990,500	90.510.410,800	0,204
720.429.080,600	710.030.130,600	4,800
5.803.212.099,000	5.810.091.386,000	23,259
45.900.140.990,000	45.900.510.780,200	146,111
720.490.000.031,000	720.502.000.000,000	420,397

Fatoração LU

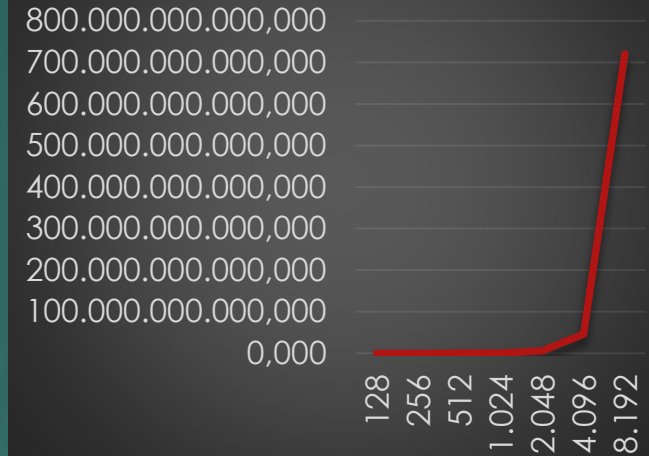
Tráfego de Dados



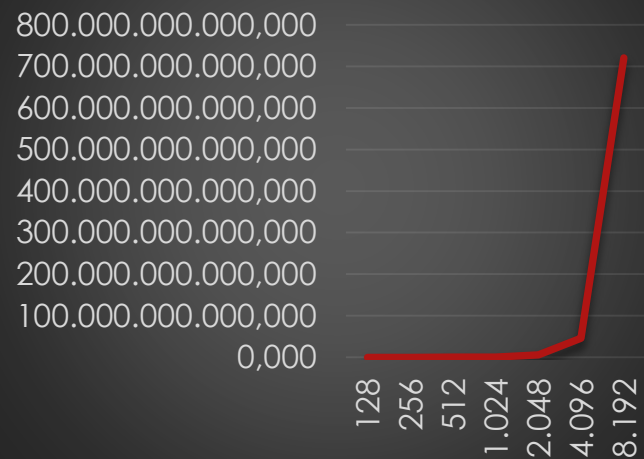
Ciclos



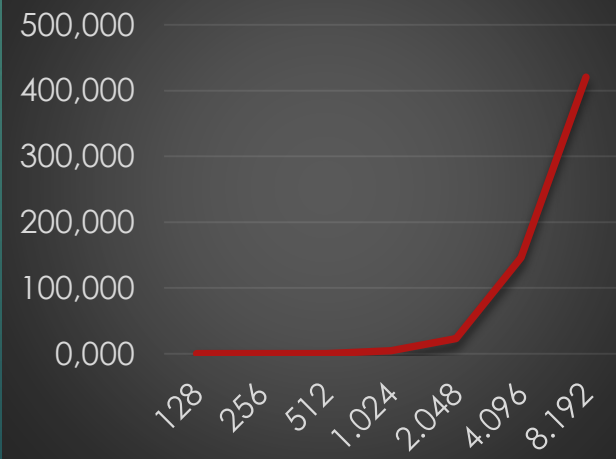
Flips



Flops



Tempo



Fatoração LU

Entre os dois algoritmos que envolvem cálculo matricial, este é o mais rápido.

Percebe-se pelo gráfico do tempo uma situação similar a da multiplicação de matrizes. Porém, encontramos um aumento significativo do tempo somente a partir de ordens após 2048. O aumento exponencial só ocorre um pouco após a matriz ultrapassar a ordem 4096.

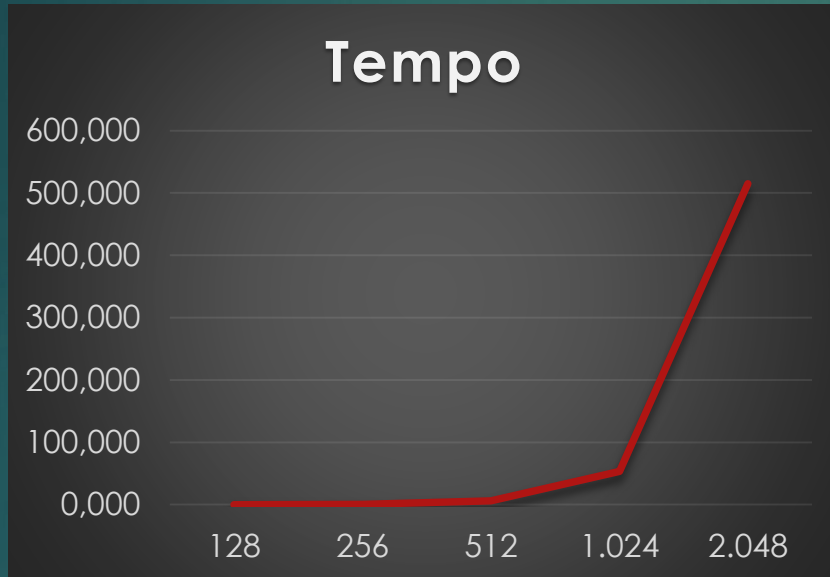
Os gráficos do total de ciclos de clock gastos, dos flips e dos flops também seguem os padrões de aumento do gráfico do tempo, ou seja, também têm os picos verticais quando a matriz atinge as mesmas ordens.

O tráfego de dados também se comporta de maneira similar. Porém, agora a diminuição drástica do seu valor ocorre até a ordem 512. E segue diminuindo cada vez mais lentamente até a ordem 1024. Após esta, o tráfego de dados passa a não possuir grande discrepância.

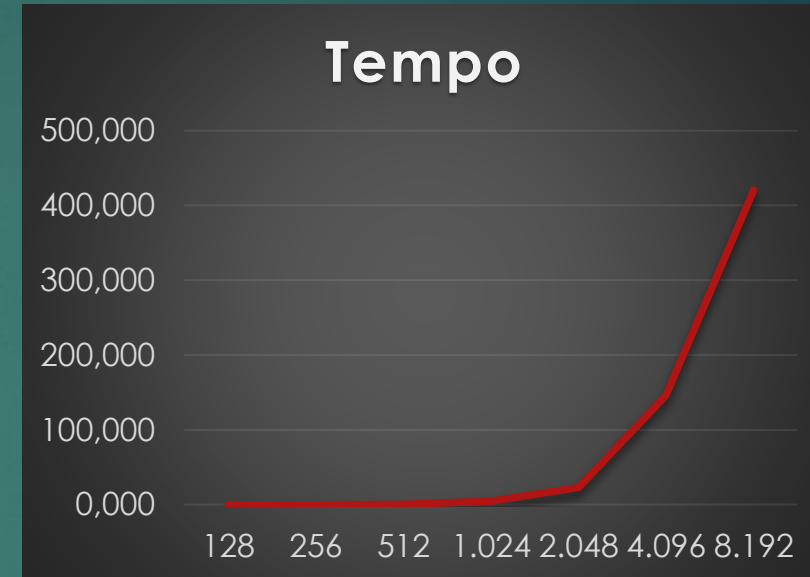
Vale ressaltar que apesar das situações serem similares entre os dois algoritmos já apresentados. A diferença de valores ápices entre os dois algoritmos anteriores altera significativamente no tempo de execução do algoritmo, como podemos observar na relação a seguir:

Comparação Matrizes x LU

Multiplicação de Matrizes



Fatoração LU



Observe que, com 400,000 segundos o algoritmo de Multiplicação de Matrizes realiza o cálculo de matrizes 2048 x 2048, aproximadamente. Enquanto o algoritmo de Fatoração LU, nesse mesmo intervalo de tempo, realiza o cálculo de matrizes 8192 x 8192.

Heap Sort

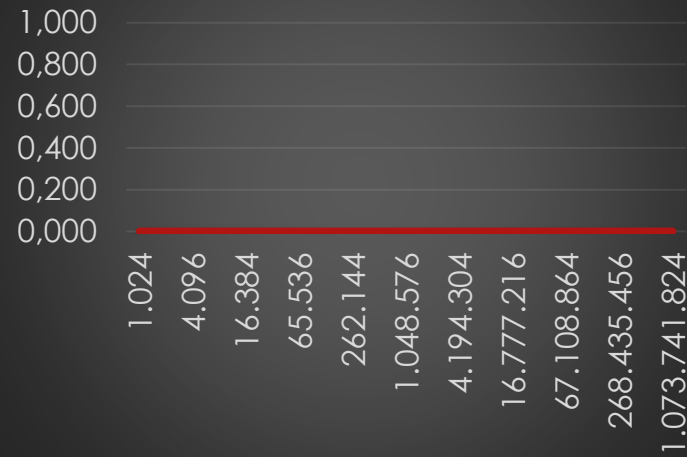
Bytes	Ciclos	Flips	Flops	Tráfego	Tempo
1.024	23.750,800	0,000	0,000	91.694.544,371	0,000
2.048	48.942,200	0,000	0,000	85.021.679,267	0,000
4.096	103.511,400	0,000	0,000	83.257.471,990	0,000
8.192	222.290,400	0,000	0,000	77.513.658,191	0,000
16.384	480.414,700	0,000	0,000	72.308.162,980	0,000
32.768	1.043.384,000	0,000	0,000	66.092.194,751	0,000
65.536	2.311.875,900	0,000	0,000	59.740.928,780	0,001
131.072	5.069.226,700	0,000	0,000	54.909.264,564	0,002
262.144	11.007.410,100	0,000	0,000	50.513.648,465	0,005
524.288	24.835.739,400	0,000	0,000	45.497.422,367	0,012
1.048.576	61.215.737,500	0,000	0,000	37.125.977,676	0,028
2.097.152	163.747.424,300	0,000	0,000	27.701.749,930	0,076
4.194.304	410.489.187,400	0,000	0,000	22.235.920,891	0,189
8.388.608	1.038.659.033,500	0,000	0,000	17.592.277,662	0,477
16.777.216	2.446.741.029,900	0,000	0,000	14.923.353,364	1,124
33.554.432	5.182.408.676,300	0,000	0,000	14.122.741,941	2,376
67.108.864	10.496.207.677,800	0,000	0,000	13.934.143,507	4,816
134.217.728	21.262.655.540,300	0,000	0,000	13.762.911,635	9,752
268.435.456	43.108.838.841,000	0,000	0,000	13.568.410,524	19,784
536.870.912	83.799.176.665,364	0,000	0,000	13.341.656,788	38,444
1.073.741.824	177.514.367.393,100	0,000	0,000	13.188.529,833	81,415

Heap Sort

Ciclos



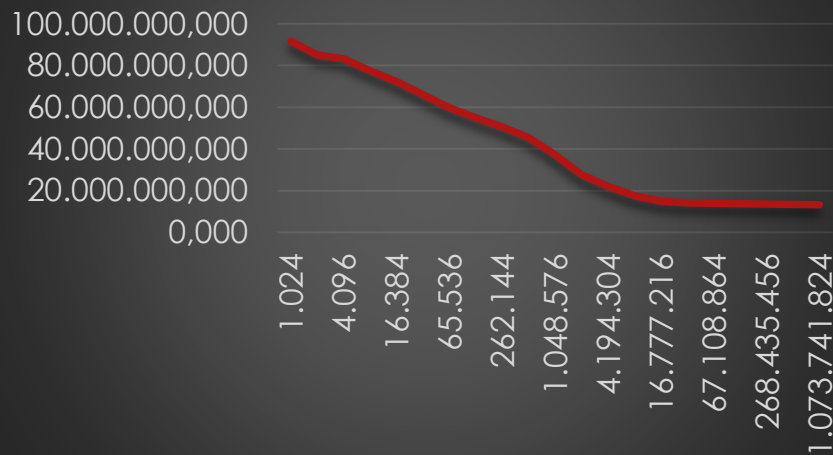
Flips



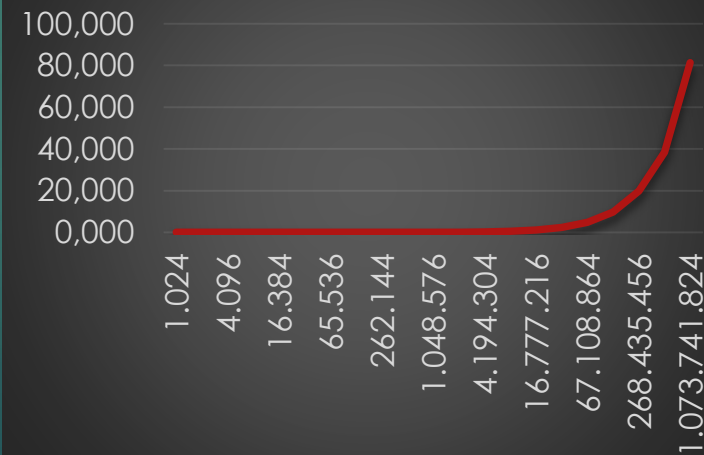
Flops



Tráfego de Dados



Tempo



Heap Sort

Entre os dois algoritmos que envolvem ordenação de vetores, este é o mais rápido, sendo também, o mais rápido dos quatro algoritmos que serão apresentados. Sua complexidade é $O(n * \log n)$.

Percebe-se pelo gráfico do tempo que o aumento significativo dos valores ocorre apenas quando o vetor atinge o tamanho de 67.108.864 bytes. Após este, o tempo aumenta gradativamente e, apenas depois do vetor ultrapassar 268.435.456 bytes, começa a aumentar exponencialmente.

Os flips e flops, porém, são 0 e permanecem assim, já que neste algoritmo, como é uma ordenação vetorial, não temos instruções e operações de pontos flutuantes.

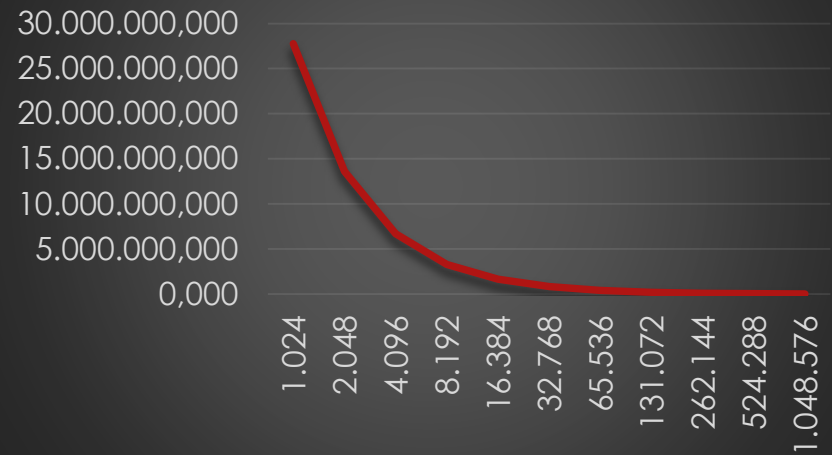
O tráfego de dados decresce muito lentamente se comparado aos dois primeiros algoritmos testados, porém também tende a ser estável quando o vetor atinge o tamanho de 16.777.216 bytes.

Bubble Sort

Bytes	Ciclos	Flips	Flops	Tráfego	Tempo
1.024	23.750,800	0,000	0,000	91.694.544,371	0,000
2.048	48.942,200	0,000	0,000	85.021.679,267	0,000
4.096	103.511,400	0,000	0,000	83.257.471,990	0,000
8.192	222.290,400	0,000	0,000	77.513.658,191	0,000
16.384	480.414,700	0,000	0,000	72.308.162,980	0,000
32.768	1.043.384,000	0,000	0,000	66.092.194,751	0,000
65.536	2.311.875,900	0,000	0,000	59.740.928,780	0,001
131.072	5.069.226,700	0,000	0,000	54.909.264,564	0,002
262.144	11.007.410,100	0,000	0,000	50.513.648,465	0,005
524.288	24.835.739,400	0,000	0,000	45.497.422,367	0,012
1.048.576	61.215.737,500	0,000	0,000	37.125.977,676	0,028
2.097.152	163.747.424,300	0,000	0,000	27.701.749,930	0,076
4.194.304	410.489.187,400	0,000	0,000	22.235.920,891	0,189
8.388.608	1.038.659.033,500	0,000	0,000	17.592.277,662	0,477
16.777.216	2.446.741.029,900	0,000	0,000	14.923.353,364	1,124
33.554.432	5.182.408.676,300	0,000	0,000	14.122.741,941	2,376
67.108.864	10.496.207.677,800	0,000	0,000	13.934.143,507	4,816
134.217.728	21.262.655.540,300	0,000	0,000	13.762.911,635	9,752
268.435.456	43.108.838.841,000	0,000	0,000	13.568.410,524	19,784
536.870.912	83.799.176.665,364	0,000	0,000	13.341.656,788	38,444
1.073.741.824	177.514.367.393,100	0,000	0,000	13.188.529,833	81,415

Bubble Sort

Tráfego de Dados



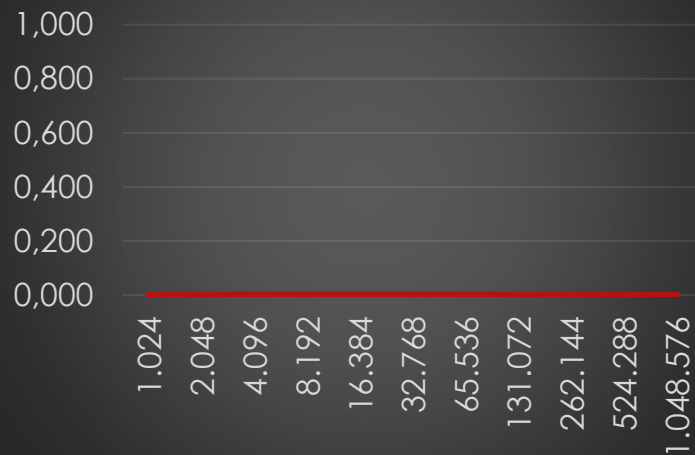
Ciclos



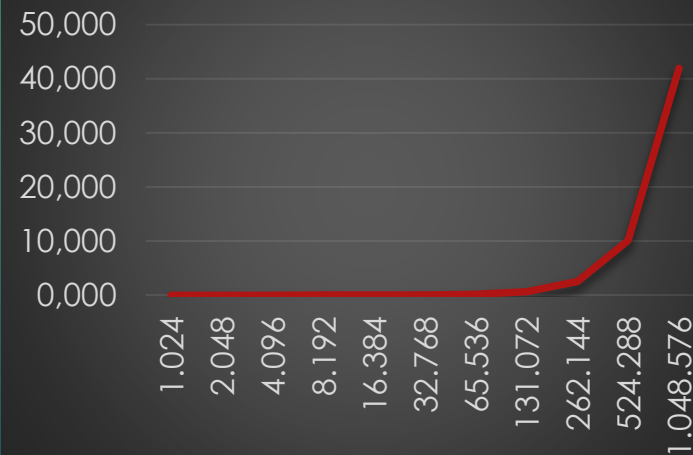
Flips



Flops



Tempo



Bubble Sort



Entre os dois algoritmos que envolvem ordenação de vetores, este é o mais lento. Sua complexidade é $O(n^2)$.

Percebe-se pelo gráfico do tempo que o aumento significativo dos valores ocorre quando o vetor atinge o tamanho de 262.144 bytes. Após este, o tempo aumenta gradativamente e, depois do vetor ultrapassar 524.288 bytes, começa a aumentar exponencialmente.

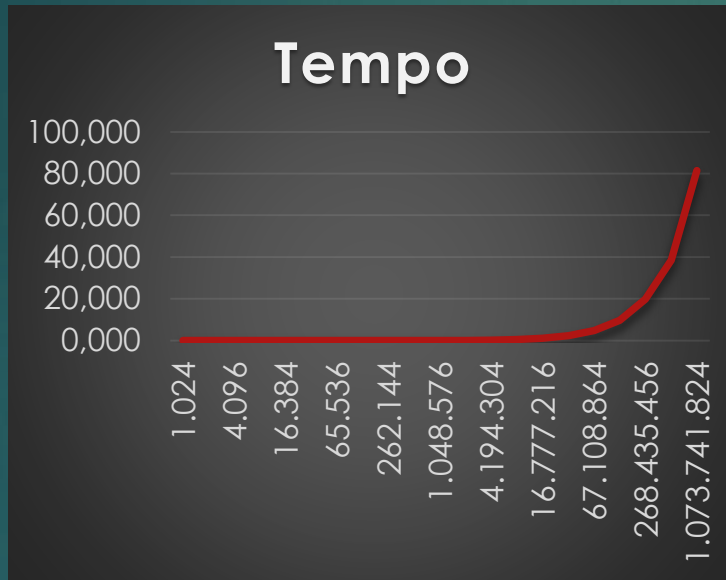
Os flips e flops também são 0 e permanecem assim pela mesma justificativa do algoritmo anterior: a ausência de instruções e operações de pontos flutuantes.

O tráfego de dados decresce rapidamente até o vetor atingir o tamanho de 2.048 bytes. Após isso, passa a decrescer cada vez mais lentamente até tender a ser estável. Isto ocorre quando o vetor atinge o tamanho de 32.768 bytes.

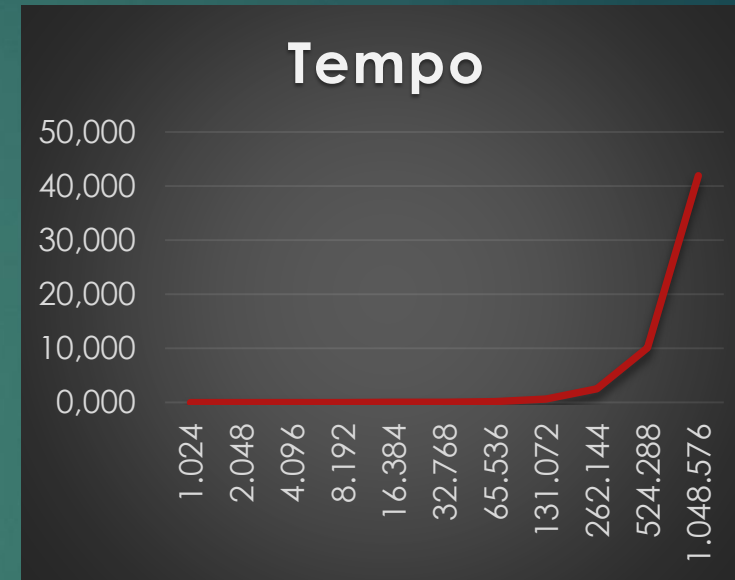
O algoritmo do Bubble Sort é significativamente mais lento do que o algoritmo do Heap Sort. Como podemos ver na relação a seguir:

Comparação Heap x Bubble

Heap Sort



Bubble Sort



Observe que, o algoritmo Heap Sort realiza a ordenação de um vetor com 1.048.576 bytes em um intervalo de tempo muito pequeno (a linha do gráfico permanece praticamente sobre o eixo horizontal). Enquanto o algoritmo Bubble Sort, ordena um vetor com essa mesma quantidade de bytes em um tempo de, aproximadamente, 40,000 segundos.