

Nome: Matheus Henrique Ferreira Protásio

Matrícula: 11521BCC020

Nome: Thiago Duarte Brito

Matrícula: 11611BCC019

## **Relatório de Análise de Algoritmos**

### **Multiplicação Comum x Multiplicação de Strassen**

A linguagem de programação adotada para implementação dos códigos foi Python. Neste experimento foram considerados apenas pares de matrizes  $n \times n$ , sendo  $n$  uma potência de 2.

Os resultados podem ser vistos abaixo:

Para a construção das tabelas, foram gerados pares aleatórios ( $n \times n$ ) e executadas as multiplicações (multiplicação comum e multiplicação de Strassen), com 30 repetições para cada par. (Com exceção de quando  $n$  era 512, que foram executadas 15 repetições, pois demorava bem mais do que as outras).

Exemplo: Gera o par  $16 \times 16$ , executa 30 vezes com o algoritmo comum e executa 30 vezes com o algoritmo de Strassen, daí os dados foram coletados, Média, Desvio Padrão, melhor execução e pior execução.

OBS: Tempo em segundos

Common				
n	Média de Tempo	Desvio Padrão	Melhor execução	Pior execução
8	0.0007	0.00003	0.0007	0.0008
16	0.0056	0.0003	0.0053	0.0073
32	0.0442	0.0031	0.0429	0.0606
64	0.3540	0.0089	0.3442	0.3814
128	2.8804	0.1088	2.7416	3.1522
256	24.7966	1.5359	21.8350	26.6961
512	194.3758	9.1968	175.9226	204.8108

Figura 1: Tabela com Dados relacionados à execução da multiplicação normal

Strassen				
n	Média de Tempo	Desvio Padrão	Melhor execução	Pior execução
8	0.0019	0.0002	0.0018	0.0035
16	0.0131	0.0003	0.0127	0.0142
32	0.0938	0.0058	0.0909	0.1192
64	0.6527	0.0215	0.6291	0.7169
128	4.6076	0.1298	4.4394	4.8745
256	35.5328	2.2073	31.2528	38.4223
512	236.3898	11.2688	215.7874	249.2901

Figura 2: Tabela com Dados relacionadas à execução da multiplicação de Strassen

Podemos observar pelas tabelas que para até matrizes 512x512, o algoritmo de Strassen teve um desempenho pior do que o algoritmo de multiplicação matricial clássico, em termos de tempo em que os algoritmos levaram para a execução da multiplicação. Porém o algoritmo de Strassen utiliza 7 chamadas recursivas, ao invés das 8 multiplicações matriciais comuns do algoritmo clássico, isso é muito bom quando aumentamos o número de n, pois conseguimos realizar cálculos mais complexos com menos operações.

Em nossos testes, com  $n=1024$ , o algoritmo de multiplicação matricial comum ainda obteve os resultados em menor tempo se comparado com o algoritmo de Strassen. Porém, testamos com matrizes 2048x2048 e obtemos o resultado em menos tempo pela função de Strassen, que se mostra uma função muito interessante para multiplicação de matrizes grandes.

Abaixo um gráfico que ilustra a curva de cada algoritmo, analisando a ordem (n) do par de matrizes e o tempo em que cada algoritmo levou para realizar a multiplicação.

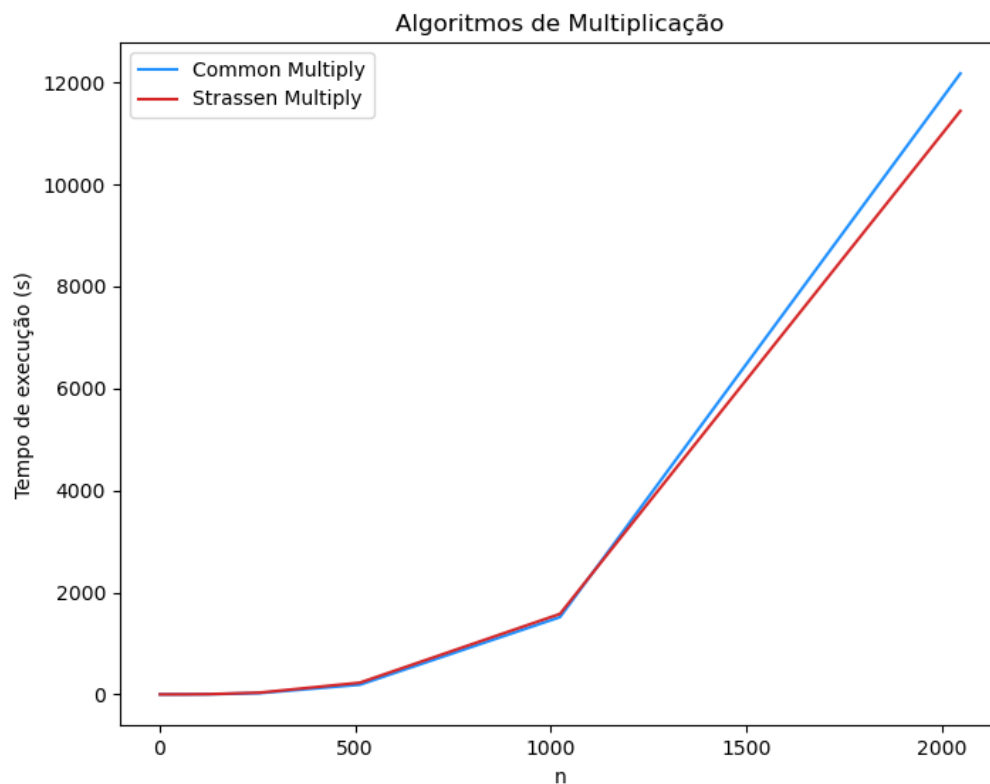


Figura 3: Gráfico Comparativo (Common x Strassen)

Utilizando os dados das tabelas e do gráfico acima, podemos observar que o algoritmo de Strassen no geral demorou mais do que o algoritmo comum de multiplicação matricial até a multiplicação de um par de matrizes  $1024 \times 1024$ . A partir disso, o algoritmo comum acaba gastando mais tempo do que o algoritmo de Strassen, que segundo testes obteve o resultado em menos tempo para a multiplicação de um par  $2048 \times 2048$ . Isso nos mostra que quanto maior a matriz, mais se tornará “inviável” o uso do algoritmo comum.

Para este relatório foram gerados cerca de 30 pares, para cada tabela tivemos cerca de 195 repetições, onde de  $n=8$  até  $n=256$  tivemos 30 repetições por algoritmo e para  $n=512$ , 15 repetições. Além disso, geramos alguns pares  $1024 \times 1024$ , mas apenas 2 pares  $2048 \times 2048$ , pois gastaram um tempo muito alto (horas) para serem executados.