

ECM306 – TÓPICOS AVANÇADOS EM ESTRUTURA DE DADOS ENGENHARIA DA COMPUTAÇÃO – 3ª SÉRIE – 2025 – Prof. Calvetti

Relatório 1 – Exercícios propostos da Aula 1

Pedro Wilian Palumbo Bevilacqua – RA: 23.01307-9

Local Experimental

Todos os experimentos e algoritmos foram rodados na mesma máquina para a diminuição de erros experimentais.

- Processador: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
- RAM instalada: 8,00 GB (utilizável: 7,73 GB)
- Tipo de sistema: Sistema operacional de 64 bits, processador baseado em x64
- Edição: Windows 11 Home Single Language
- Versão: 23H2 - Instalado em 19/07/2023
- Compilação do SO: 22631.4890
- Armazenamento: SSD de 256GB

Resultados dos Exercícios

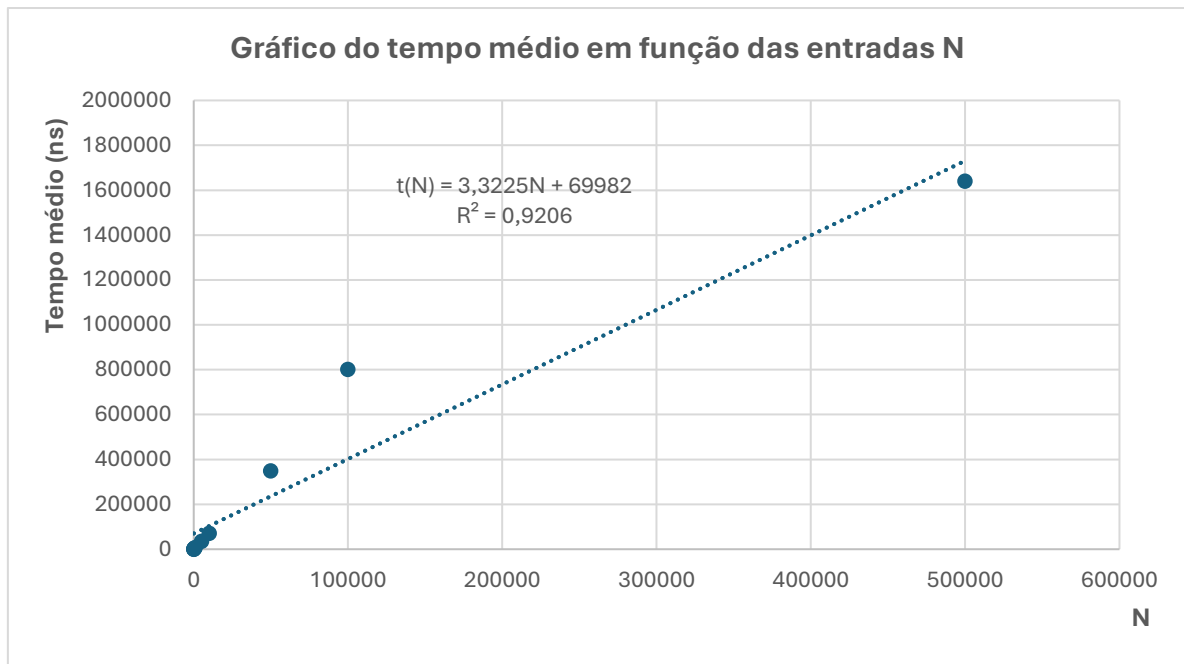
Exercício 1 - Elaborar um programa, em linguagem Java, capaz inicializar com 0 (zero) um vetor do tipo int de N elementos, utilizando laços, onde N deve ser: 10, 50, 100, 500, 1000, 5000, 10000, 50000, 100000 e 500000.

a) Medição do tempo de execução do algoritmo em nanosegundos:

Tabela dos dados coletados:

N	10	50	100	500	1000	5000	10000	50000	100000	500000
T1 (ns)	600	1100	1400	6000	7400	34300	67600	369500	757300	1898700
T2 (ns)	500	700	1200	4000	7300	34600	73200	332300	882300	1602700
T3 (ns)	600	1200	1100	4000	6800	37500	70000	366600	869800	1610800
T4 (ns)	400	1200	1200	5600	7400	37200	73000	338200	751300	1533800
T5 (ns)	400	800	1600	5800	7400	34600	72900	338800	750900	1560500
Tmédio (ns)	500	1000	1300	5080	7260	35640	71340	349080	802320	1641300

Gráfico dos dados da tabela coletados:



A partir do gráfico podemos identificar que o algoritmo possui tendência linear, podendo ser observado pelo R^2 .

b) Medição das Operações Aritméticas e Comparações:

Tabela da contagem obtida pelos contadores do código:

N	10	50	100	500	1000	5000	10000	50000	100000	500000
Comparações	11	51	101	501	1001	5001	10001	50001	100001	500001
Operações Aritmética	10	50	100	500	1000	5000	10000	50000	100000	500000

Além da tabela, podemos deduzir as expressões gerais de ambas as comparações e operações aritméticas:

Comparação: $n = N$

Operações: $n = N + 1$

Exercício 2 - Elaborar um programa, em linguagem Java, capaz copiar o conteúdo de um vetor do tipo int de N elementos para outro vetor de mesmo tamanho, utilizando laços, onde N deve ser: 10, 50, 100, 500, 1000, 5000, 10000, 50000, 100000 e 500000.

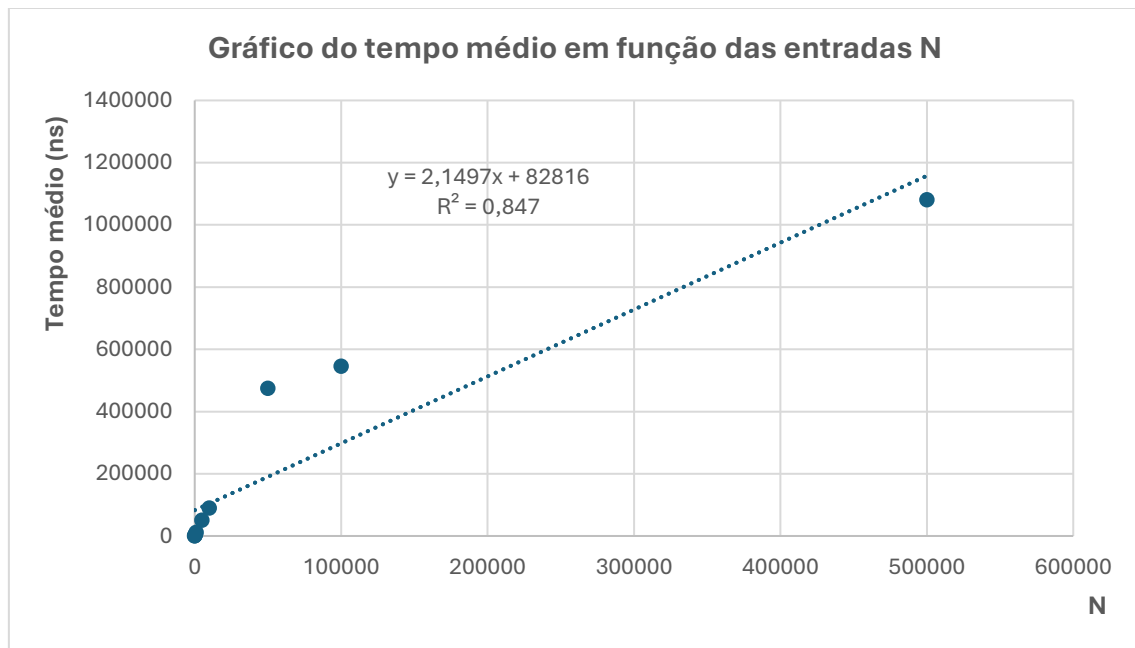
a) Medição do tempo de execução do algoritmo em nanosegundos:

Tabela dos dados coletados:

N	10	50	100	500	1000	5000	10000	50000	100000	500000
T1 (ns)	500	1000	1600	4100	13000	43400	85700	454500	861500	1356000

T2 (ns)	500	1000	1400	4500	10000	62100	83900	432500	221700	1057900
T3 (ns)	500	800	1600	6000	10100	43100	108100	427900	508200	993500
T4 (ns)	800	800	1100	4300	10100	43200	93000	449400	559400	1029600
T5 (ns)	500	800	1200	5800	14000	63800	78900	611000	575900	966300
Tmédio (ns)	560	880	1380	4940	11440	51120	89920	475060	545340	1080660

Gráfico dos dados da tabela coletados:



A partir do gráfico podemos identificar que o algoritmo possui tendência linear, podendo ser observado pelo R^2 .

b) Medição das Operações Aritméticas e Comparações:

Tabela da contagem obtida pelos contadores do código:

N	10	50	100	500	1000	5000	10000	50000	100000	500000
Comparações	11	51	101	501	1001	5001	10001	50001	100001	500001
Operações Aritmética	10	50	100	500	1000	5000	10000	50000	100000	500000

Além da tabela, podemos deduzir as expressões gerais de ambas as comparações e operações aritméticas:

Comparação: $n = N$

Operações: $n = N+1$

Exercício 3 - Elaborar um programa, em linguagem Java, capaz de limpar (colocar -1 em todas as posições) matrizes de dimensões N por M, do tipo double, onde N e M devem ser, respectivamente: 10 e 10; 50 e 75; 100 e 300; 500 e 200; 1000 e 1000; 5000 e 7000; 10000 e 1; 50000 e 25000; 100000 e 100000; e 500000 e 1000.

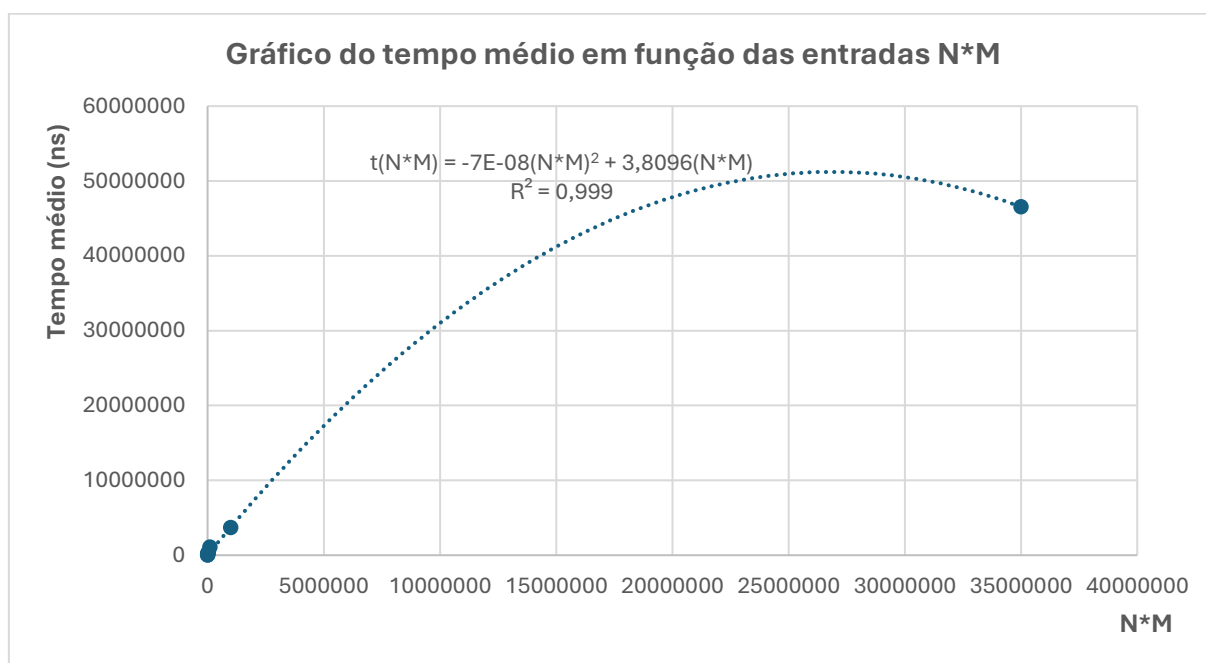
a) Medição do tempo de execução do algoritmo em nanosegundos:

Tabela dos dados coletados:

N	10	50	100	500	1000	5000	10000	50000	100000	500000
M	10	75	300	200	1000	7000	1	25000	100000	1000
N x M	100	3750	30000	100000	1000000	35000000	10000	1,25E+09	1E+10	500000000
T1 (ns)	1600	55300	282300	1403500	5926300	47215500	301700			
T2 (ns)	1700	38000	488400	964700	2881900	46750700	182200			
T3 (ns)	1500	41100	404700	965300	2842600	45148500	182100			
T4 (ns)	1700	34200	261500	942400	3046600	48544500	190500			
T5 (ns)	2300	33600	277900	993600	3606000	45073300	176100			
Tmédio (ns)	1760	40440	342960	1053900	3660680	46546500	206520			

Os valores não digitados ou valores nulos, demonstram que não foram possível obter um resultado, porque o código da um erro “*Exception in thread "main" java.lang.OutOfMemoryError: Java heap space*”. Sendo um erro geralmente causado por falta de memória RAM para o código.

Gráfico dos dados da tabela coletados:



A partir do gráfico podemos identificar que o algoritmo possui tendência quadrática, podendo ser observado pelo R^2 .

b) Medição das Operações Aritméticas e Comparações:

Tabela da contagem obtida pelos contadores do código:

N x M	10x10	50x75	100x300	500x200	1k x 1k	5k x 7k	10k x 1	50k x 25k	100k x 100k	500k x 1k
Comparações	121	3851	30201	101001	1002001	35010001	30001			
Operações Aritmética	110	3800	30100	100500	1001000	35005000	20000			

Além da tabela, podemos deduzir a expressões gerais de ambas as comparações e operações aritméticas:

Para matrizes quadradas:

Comparação: $n = (N+1)^2$

Operações: $n = N + N^2$

Para matrizes não quadradas:

Comparação: $n = N(M+1) + (N+1)$

Operações: $n = N(M+1)$

A partir disso podemos substituir os valores nulos, pelos seus resultados analíticos usando as expressões gerais:

N x M	10x10	50x75	100x300	500x200	1k x 1k	5k x 7k	10k x 1	50k x 25k	100k x 100k	500k x 1k
Comparações	121	3851	30201	101001	1002001	35010001	30001	1,25E+09	1,00E+10	5,01E+08
Operações Aritmética	110	3800	30100	100500	1001000	35005000	20000	1,25E+09	1,00E+10	5,01E+08

Exercício 4 - Elaborar um programa, em linguagem Java, capaz de informar quando uma matriz N por N, do tipo int, é simétrica (quando a matriz analisada for igual à sua transposta), onde N deve ser: 10, 50, 100, 500, 1000, 5000, 10000, 50000, 100000 e 500000.

a) Medição do tempo de execução do algoritmo em nanosegundos:

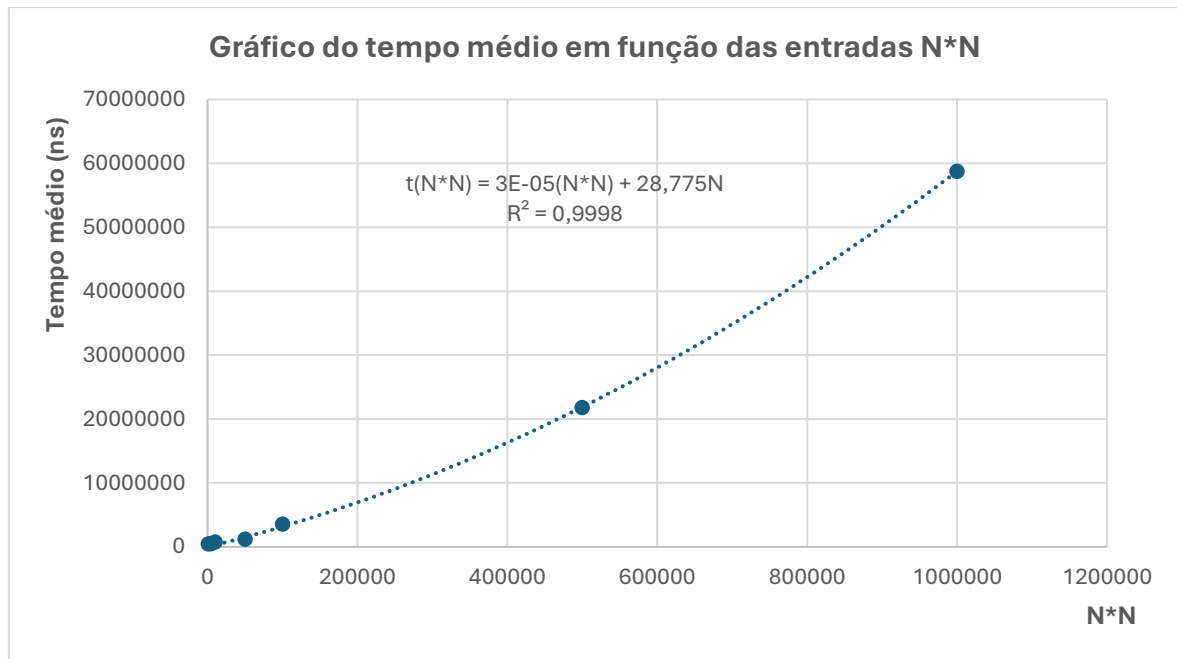
Tabela dos dados coletados:

N x M	10 ²	50 ²	100 ²	500 ²	1000 ²	5000 ²	10000 ²	50000 ²	100000 ²	500000 ²
T1 (ns)	508500	428500	831500	1390300	4192800	22562600	57392200			
T2 (ns)	539000	531200	699100	1201400	3164200	19578500	54641900			
T3 (ns)	559400	549400	918500	1229700	4024900	22846200	59569300			
T4 (ns)	366200	559800	705300	1138100	3296300	22263300	62781600			
T5 (ns)	425600	489600	574100	1144800	3087100	21646000	59576800			

Tmédio (ns)	479740	511700	745700	1220860	3553060	21779320	58792360			
-------------	--------	--------	--------	---------	---------	----------	----------	--	--	--

Os valores não digitados ou valores nulos, demonstram que não foram possível obter um resultado, porque o código da um erro “*Exception in thread "main" java.lang.OutOfMemoryError: Java heap space*”. Sendo um erro geralmente causado por falta de memória RAM para o código.

Gráfico dos dados da tabela coletados:



A partir do gráfico podemos identificar que o algoritmo possui tendência quadrática, podendo ser observado pelo R^2 .

b) Medição das Operações Aritméticas e Comparações:

Tabela da contagem obtida pelos contadores do código:

N x N	10x10	50x50	100x100	500x500	1k x 1k	5k x 5k	10k x 10k	50k x 50k	100k x 100k	500k x 500k
Comparações	222	5102	20202	501002	2002002	50010002	2E+08			
Operações Aritmética	111	2551	10101	250501	1001001	25005001	1E+08			

Além da tabela, podemos deduzir a expressões gerais de ambas as comparações e operações aritméticas:

Comparação: $n = (N+2) + N[(N + 1)+N] =$

Operações: $n = (N+1)+ N^2$

A partir disso podemos substituir os valores nulos, pelos seus resultados analíticos usando as expressões gerais:

N x N	10x10	50x50	100x100	500x500	1k x 1k	5k x 5k	10k x 10k	50k x 50k	100k x 100k	500k x 500k
Comparações	222	5102	20202	501002	2002002	50010002	2E+08	5000100002	2,00E+10	5,00E+11
Operações Aritmética	111	2551	10101	250501	1001001	25005001	1E+08	2500050001	1,00E+10	2,50E+11