

ECM306 – TÓPICOS AVANÇADOS EM ESTRUTURA DE DADOS

ENGENHARIA DA COMPUTAÇÃO – 3ª SÉRIE – 2024 – E1, E2

LAB – PROF. CALVETTI

EXERCÍCIOS PROPOSTOS – AULA 01

Aluna: Amanda Carolina Ambrizzi Ramin; R.A: 22.00721-0

Exercício 1:

Código:

```
public class Ex1
{
    public static void main(String[] args) {
        int N[] = {10, 50, 100, 500, 1000, 5000, 10000, 50000, 100000,
500000};

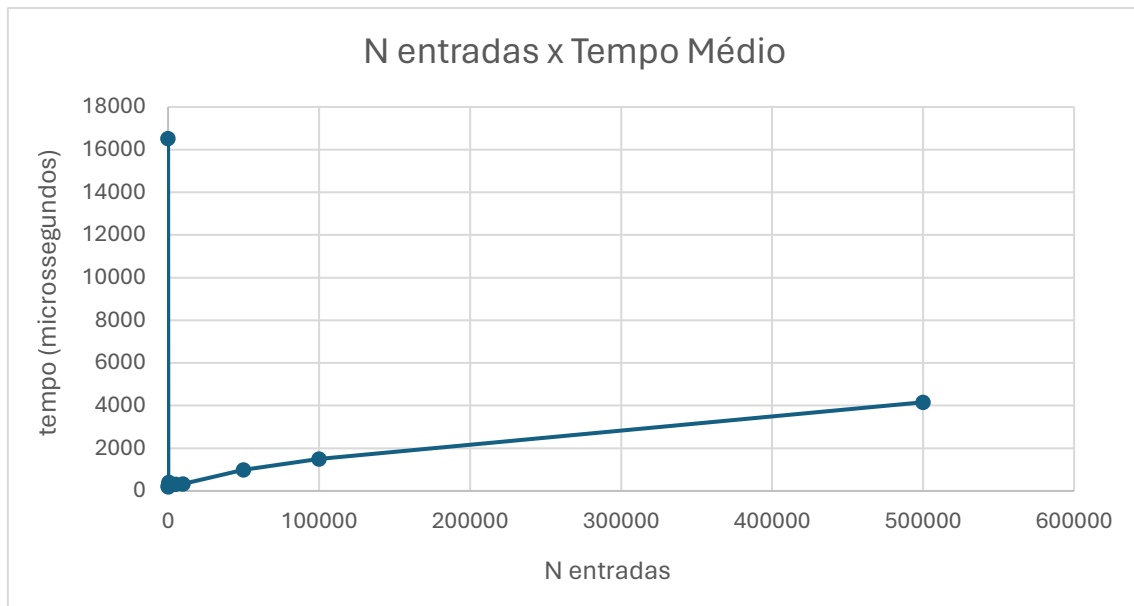
        for(int x : N){
            int vetor[] = new int[x];
            for(int i = 0; i < x; i++){
                vetor[i] = 0;
            }
            System.out.println("Vetor de " + x + " posicoes criado.");
        }
    }
}
```

Dados Coletados:

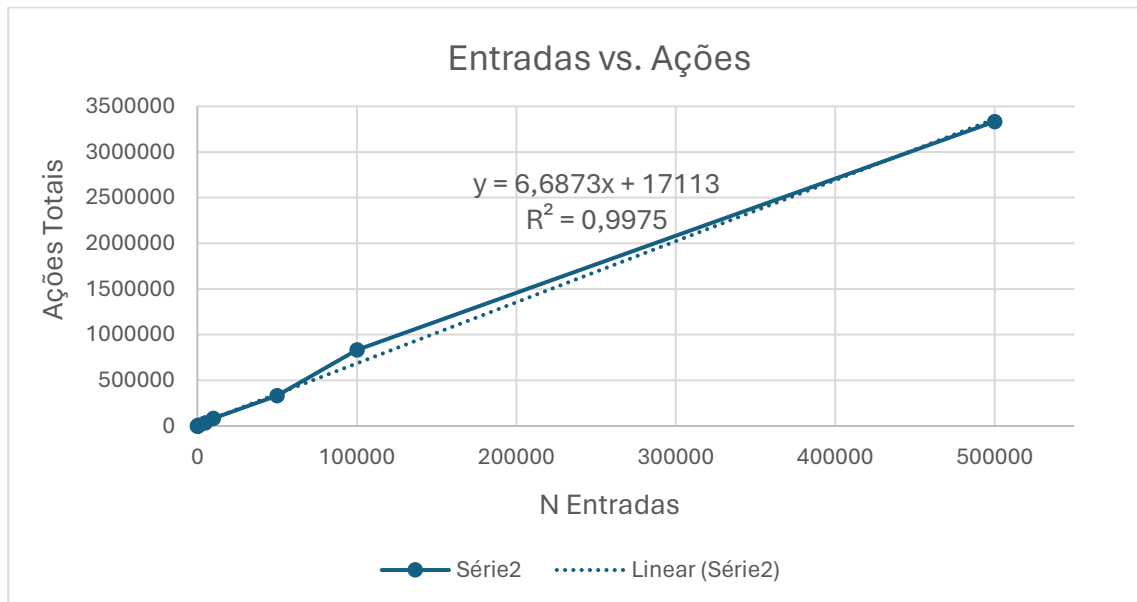
i	Entradas (N)	Tempos 1	Tempos 2	Tempos 3	Tempos 4	Tempos 5	Tempo médio	Comparações	Aritmética	Atribuições	Ações Totais
0	10	18143	15045	16607	15208	17586	16517,8	10	10	33	53
1	50	352	70	192	151	208	194,6	60	60	185	305
2	100	204	75	153	481	271	236,8	160	160	487	807
3	500	204	169	208	255	1194	406	660	660	1898	3218
4	1000	228	187	268	299	410	278,4	1660	1660	4991	8311
5	5000	212	322	309	276	440	311,8	6660	6660	19993	33313
6	10000	248	463	370	265	292	327,6	16660	16660	49995	83315
7	50000	875	1574	895	926	705	995	66660	66660	199997	333317
8	100000	1467	1524	1478	1654	1384	1501,4	166660	166660	499999	833319
9	500000	3020	3658	4815	3246	6060	4159,8	666660	666660	2000001	3333321

*tempo medido em microssegundos

Análise por Tempo de Execução:



Análise por Operações Aritméticas, Atribuições e Comparações:



Exercício 2:

Código:

```
public class Ex2
{
    public static void main(String[] args) {
        int N[] = {10, 50, 100, 500, 1000, 5000, 10000, 50000, 100000, 500000};

        for(int x : N){

            int vetor_original[] = new int[x];
            int vetor_copia[] = new int[x];
```

```

    for(int i = 0; i < x; i++){
        vetor_original[i] = 0;
    }

    for(int j = 0; j < x; j ++){
        vetor_copia[j] = vetor_original[j];
    }

    System.out.println("Vetor de " + x + " posicoes copiado.");
}
}
}

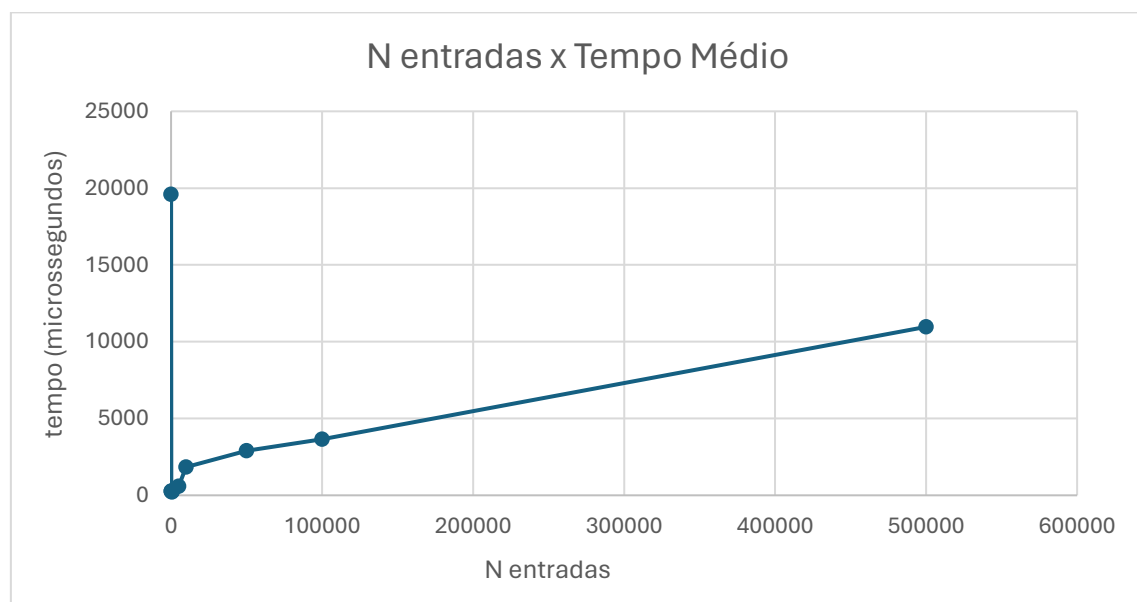
```

Dados Coletados:

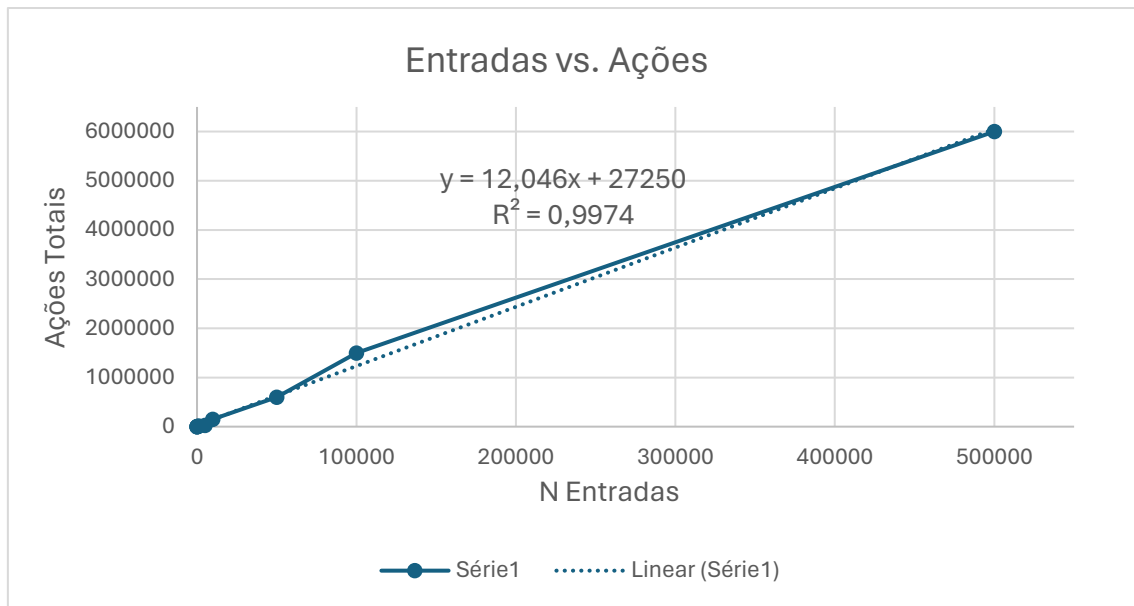
i	Entradas (N)	Tempos 1	Tempos 2	Tempos 3	Tempos 4	Tempos 5	Tempo médio	Comparações	Aritmética	Atribuições	Ações Totais
0	10	32063	17778	17966	17266	12933	19601,2	20	20	54	94
1	50	607	69	160	204	178	243,6	120	120	306	546
2	100	524	59	440	225	145	278,6	320	320	808	1448
3	500	293	171	227	246	164	220,2	1320	1320	3310	5950
4	1000	375	239	282	228	228	270,4	3320	3320	8312	14952
5	5000	1013	359	750	485	371	595,6	13320	13320	3314	29954
6	10000	1337	823	6112	478	487	1847,4	33320	33320	83316	149956
7	50000	5101	2879	2658	1490	2356	2896,8	133320	133320	333318	599958
8	100000	4215	4001	2609	3314	4071	3642	333320	333320	833320	1499960
9	500000	29532	5111	6091	7096	7019	10969,8	1333320	1333320	3333322	5999962

*tempo medido em microssegundos

Tempo de Execução:



Operações Aritméticas, Atribuições e Comparações:



Exercício 3:

Código:

```
public class Ex3
{
    public static void main(String[] args) {
        //int N[] = {10, 50, 100, 500, 1000, 5000, 10000, 50000, 100000,
        500000};
        //int M[] = {10, 75, 300, 200, 1000, 7000, 1, 25000, 100000,
        1000};

        /*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 */

        int N = 10;
        int M = 10;
        double matriz[][] = new double[N][M];

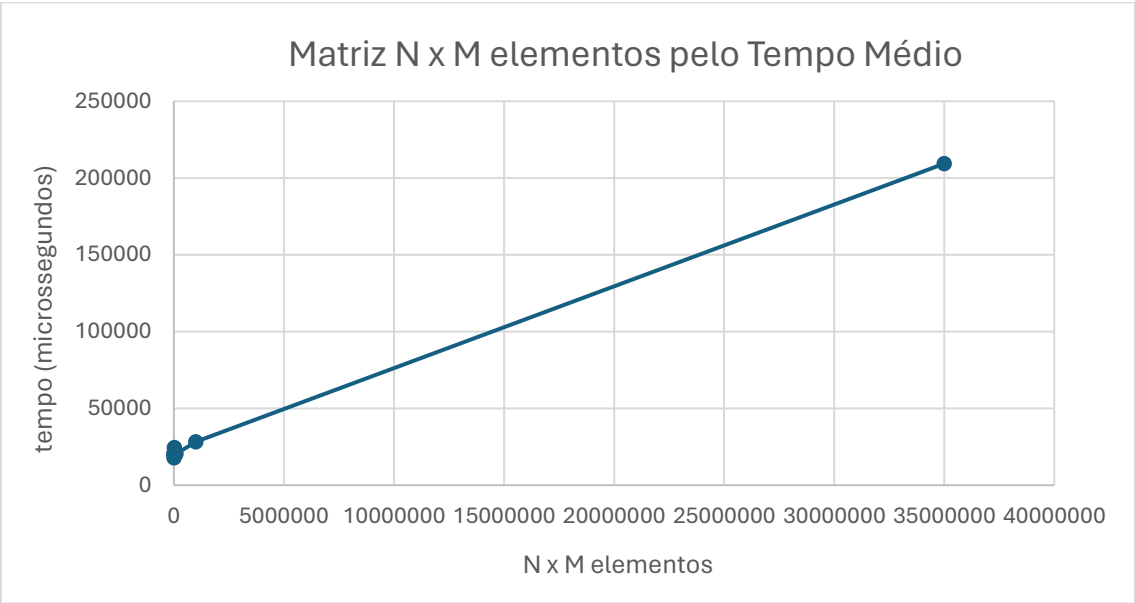
        for(int i = 0; i < N; i++){
            for(int j = 0; j < matriz[i].length; j++){
                matriz[i][j] = -1;
            }
        }
        System.out.println("Matriz de " + N + " x " + M + " foi limpa.");
    }
}
```

Dados Coletados:

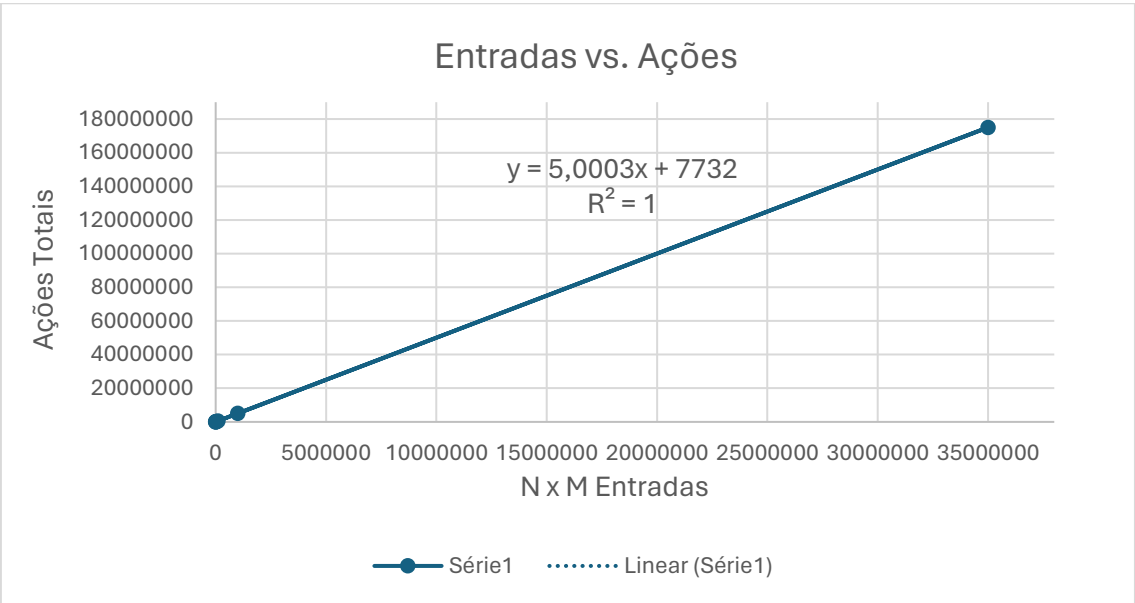
i	Entradas (N)	Entrada(M)	Tempos 1	Tempos 2	Tempos 3	Tempos 4	Tempos 5	Tempo médio	Comparações	Aritmética	Atribuições	M*N	Ações Totais
0	10	10	22745	20271	17403	18847	21917	20236,6	110	110	323	100	543
1	50	75	24237	22567	22315	16708	16531	20471,6	3800	3800	11353	3750	18953
2	100	300	22001	32304	20967	16296	31233	24560,2	30100	30100	90203	30000	150403
3	500	200	22000	19728	17826	19500	22871	20385	100500	100500	301003	100000	502003
4	1000	1000	28230	28951	27184	25508	31267	28228	1001000	1001000	3002003	1000000	5004003
5	5000	7000	231289	193712	212272	216911	192567	209350,2	35005000	35005000	105010003	35000000	175020003
6	10000	1	17329	16664	16761	16678	22004	17887,2	20000	20000	50003	10000	90003
7	50000	25000	-	-	-	-	-	-	-	-	-	-	-
8	100000	100000	-	-	-	-	-	-	-	-	-	-	-
9	500000	1000	-	-	-	-	-	-	-	-	-	-	-

*tempo medido em microssegundos

Tempo de Execução:



Operações Aritméticas, Atribuições e Comparações:



Exercício 4:

Código:

```
public class Ex4 {
    public static void main(String[] args) {
        //int N[] = {10, 50, 100, 500, 1000, 5000, 10000, 50000, 100000,
        500000};

        int N = 10;
        boolean simetria = true;

        int matriz[][] = new int[N][N];
        for(int i = 0; i < N; i++){
            for(int j = 0; j < N; j++){
                matriz[i][j] = 0;
            }
        }

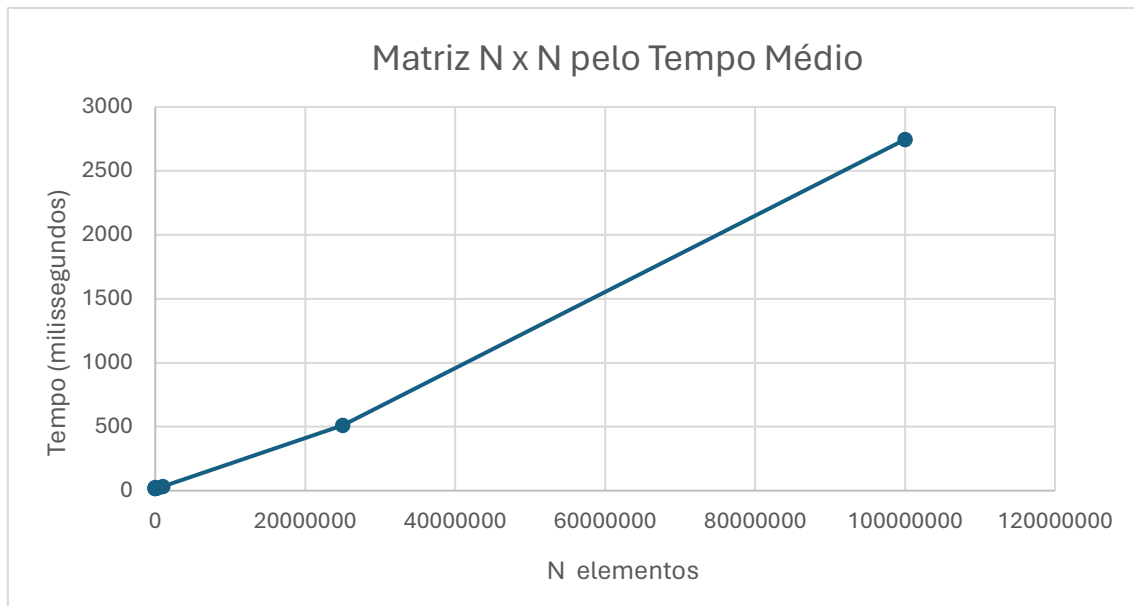
        for(int i = 0; i < N; i++){
            for(int j = 0; j < N; j++){
                if(matriz[i][j] != matriz[j][i]){
                    simetria = false;
                }
            }
        }
        System.out.println("Simétrica? " + simetria);
    }
}
```

Dados Coletados:

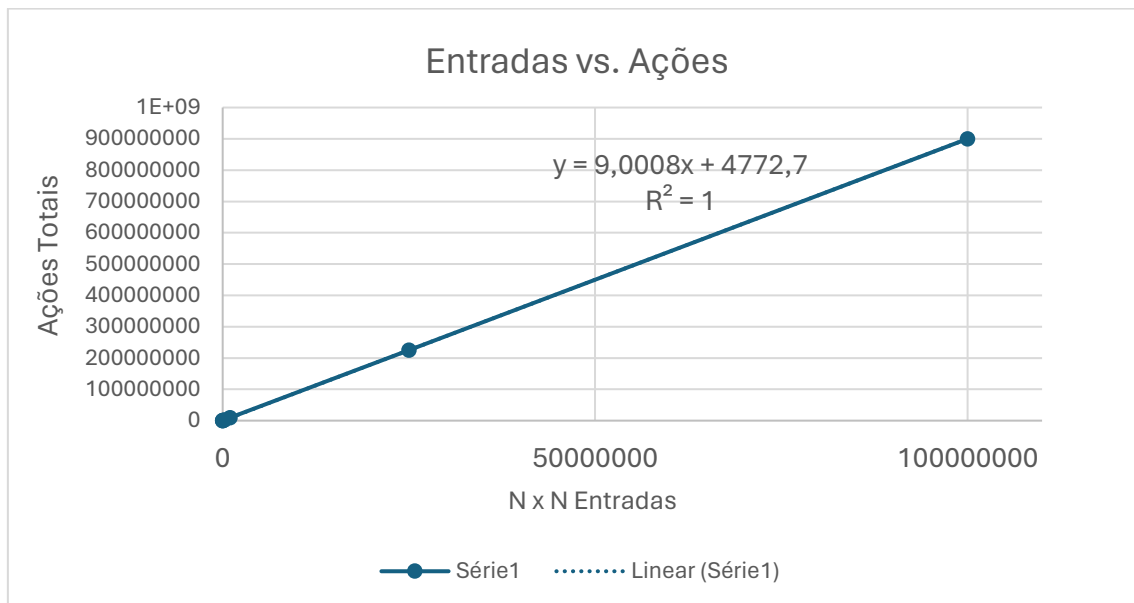
i	Entradas (N)	Tempos 1	Tempos 2	Tempos 3	Tempos 4	Tempos 5	Tempo médio	Comparações	Aritmética	Atribuições	N x N	Ações Totais
0	10	15	18	62	15	17	25,4	220	220	543	100	983
1	50	17	21	15	15	17	17	5100	5100	12703	2500	22903
2	100	18	16	19	17	14	16,8	20200	20200	50403	10000	90803
3	500	20	21	25	17	21	20,8	501000	501000	1252003	250000	2254003
4	1000	29	27	34	43	35	33,6	2002000	2002000	5004003	1000000	9008003
5	5000	522	489	519	463	556	509,8	50010000	50010000	125020003	25000000	225040003
6	10000	3154	2889	2546	2600	2547	2747,2	200020000	200020000	500040003	100000000	900080003
7	50000	-	-	-	-	-	-	-	-	-	2500000000	-
8	100000	-	-	-	-	-	-	-	-	-	10000000000	-
9	500000	-	-	-	-	-	-	-	-	-	2,5E+11	-

*tempo medido em milissegundos

Tempo de Execução:



Operações Aritméticas, Atribuições e Comparações:



Análise Geral pelo Tempo de Execução:

Foi possível observar pelos gráficos dos exercícios 1 e 2 que a máquina usada para rodar o algoritmo desenvolvido demorou para alocar a memória sequencial necessária para iniciar o primeiro vetor. Porém, para os demais vetores, esse processo foi mais rápido e o comportamento mostrou que o aumento do número de entradas ocasiona o aumento do tempo de execução.

Já para os algoritmos dos exercícios 3 e 4, além da relação entrada vs. tempo citada acima, nota-se que os dados a partir de $N = 50000$ não puderam ser coletados devido à falta de memória sequencial do computador para criar as matrizes.

Análise Geral por Operações Aritméticas, Atribuições e Comparações:

A partir das tabelas de quantidade de operações aritméticas, atribuições e comparações, pode-se afirmar que as ações aumentaram conforme o número de variáveis. Sendo que as comparações e operações aritméticas tiveram valores iguais e mais baixos quando comparado a quantidade de atribuições. Além disso, existe uma relação de linearidade enquanto ao número de entradas e a os parâmetros analisados que todo ser visto nos gráficos de Ações por Entrada de cada exercício.

Conclusão

Como conclusão geral temos que utilizar apenas um dos métodos de análise não é suficiente para concluir a eficiência de um algoritmo. Além disso, existem fatores que influenciam nos resultados, como o processador do computador e as outras instâncias em execução, que podem prejudicar o funcionamento do programa.