

Algoritmo Faz Algo

DCC606 - Análise de Algoritmos

Aluno: Guilherme Lucas Pereira Bernardo
- 2019004044 -



Código em C:



```
void FazAlgo(int n) {  
    int i, j, k;  
    int controle = 0;  
    for (i = 1; i < n - 1; i++) {  
        for (j = i + 1; j ≤ n; j++) {  
            for (k = 1; k ≤ j; k++) {  
                controle++;  
            }  
        }  
    }  
}
```

Testes de performance

Resultados dos testes(média de 13 testes) em segundos:

entradas	tempo de conclusão
1	0
100	0.00053
200	0.00417
300	0.01391
400	0.03261
500	0.06313
600	0.10883
700	0.17272
800	0.25701
900	0.36645

entradas	tempo de conclusão
1000	0.50191
2000	4.02019
3000	13.50029
4000	31.90756
5000	61.99627
6000	107.19423
7000	170.46625
8000	253.84731
9000	361.91553
10000	494.20585

entradas	tempo de conclusão
15000	1676.00652
20000	3996.38799

Plataforma usada:

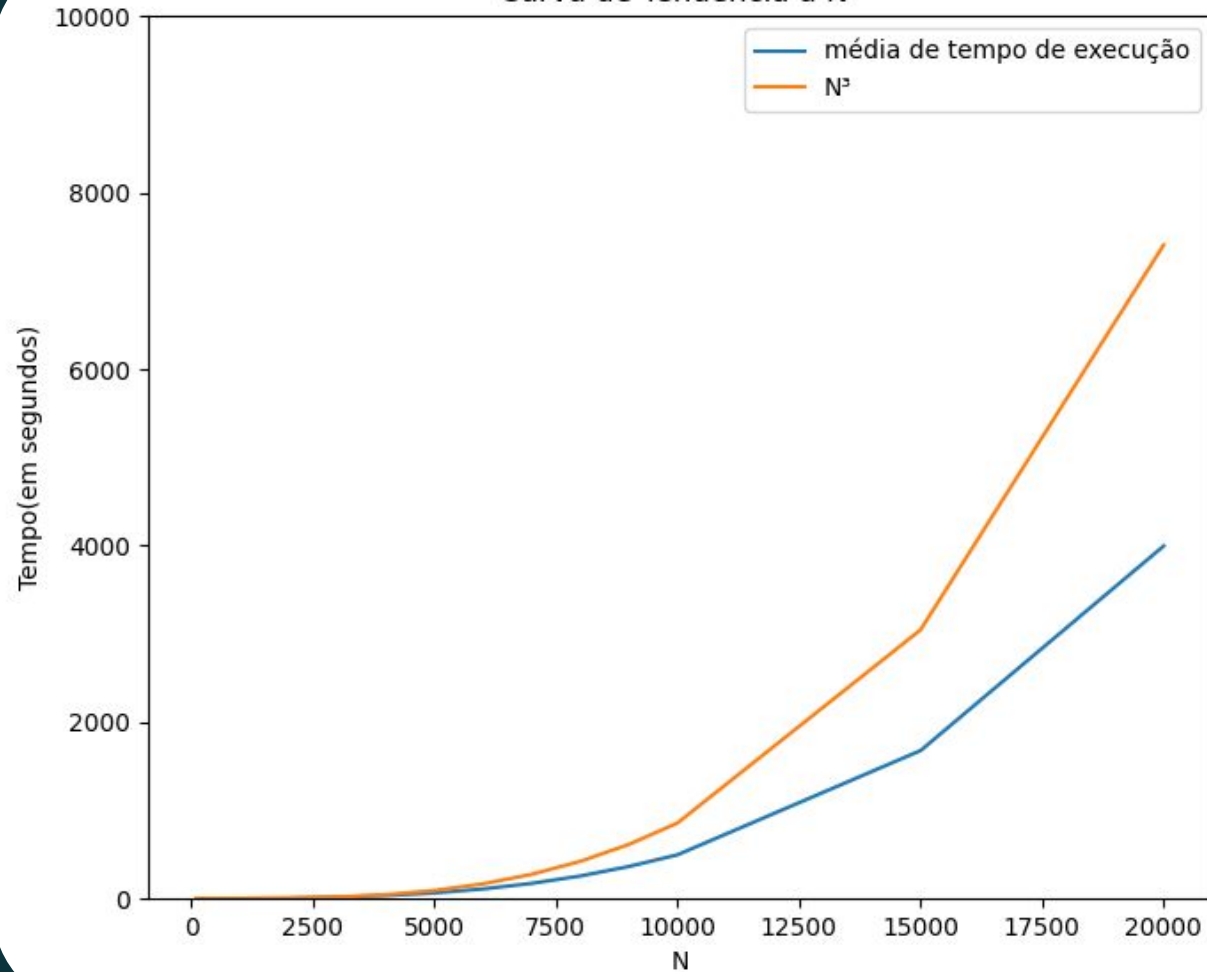
- Placa mãe: MSI MAG B550M BAZOOKA AMD AM4 DDR4.
- Memória RAM: 2x8 GB XPG Spectrix 3600MHz, DDR4, CL18.
- Processador: Ryzen 7 5700G 4.5Ghz.
- Placa de Vídeo: AMD Radeon™ Graphics.
- Sistema Operacional: Ubuntu linux (wsl mode on Windows 11 Home 22H2).
- SSD: 512GB NVME M2 Leitura 7200MB/s e Gravação 2600MB/s.



Demonstração em gráfico



Curva de Tendência a N^3



Função de custo

$$F(N) = \frac{2 N^3 - 3 N^2 + N}{2}$$

$O(n^3)$

Complexidade

Algoritmo Otimizado



Código em C:

```
void FazAlgov2(int n) {  
    int i, j, k;  
    int controle = 0;  
    for (i = 1; i < n - 1; i++) {  
        for (j = i + 1; j ≤ n; j++) {  
            controle++;  
            controle = controle + (j - 1);  
        }  
    }  
}
```


Algoritmo Otimizado

Comparaão de tempos de execuão entre o algoritmo Original e o Otimizado

Algoritmo Antigo

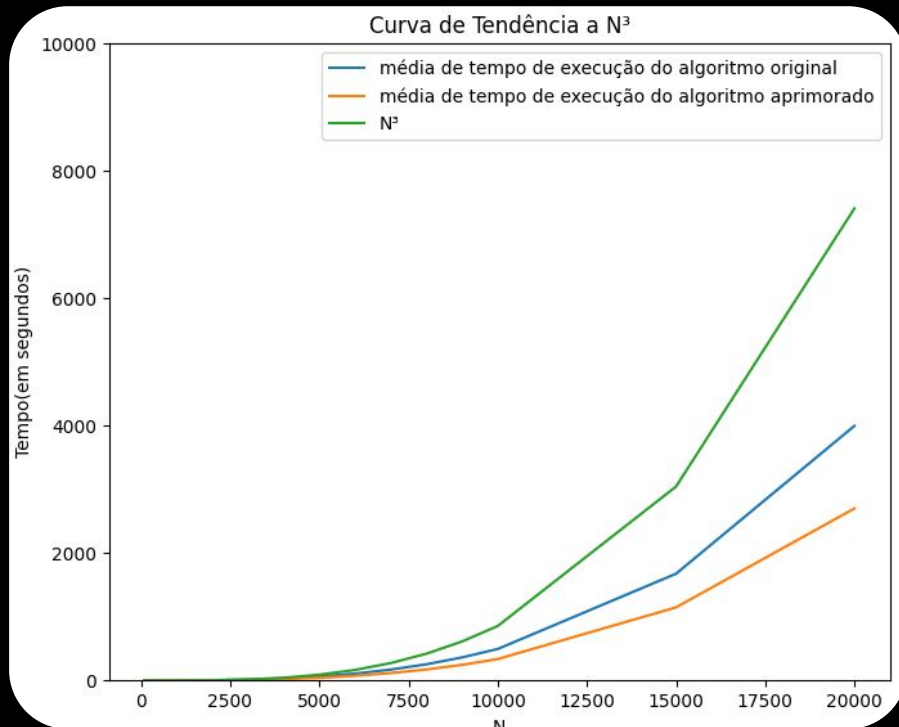
entradas	tempo de concluso	entradas	tempo de concluso
100	0.00053	3000	13.50029
200	0.00417	4000	31.90756
300	0.01391	5000	61.99627
400	0.03261	6000	107.19423
500	0.06313	7000	170.46625
600	0.10883	8000	253.84731
700	0.17272	9000	361.91553
800	0.25701	10000	494.20585
900	0.36645	15000	1676.00652
1000	0.50191	20000	3996.38799
2000	4.02019		

Algoritmo Otimizado

entradas	tempo de concluso	entradas	tempo de concluso
100	0.00037	3000	9.09363
200	0.00283	4000	21.50791
300	0.00933	5000	42.01840
400	0.02192	6000	72.92552
500	0.04563	7000	115.90335
600	0.07406	8000	172.45884
700	0.11704	9000	244.77263
800	0.17475	10000	335.80355
900	0.24806	15000	1147.38451
1000	0.33852	20000	2685.374574
2000	2.70273		

Algoritmo Otimizado

Demonstração em gráfico



	Média de tempo (todos os testes)
Algoritmo Original	341.56996
Algoritmo Aprimorado	230.99895

Com isso, é possível observar que o algoritmo Otimizado é **32,3715%** mais eficiente que o Original.



Fim de apresentação

Dúvidas?

