INSTITUTO FEDERAL
Maranhão

Estudo Comparativo do Desempenho da Busca Linear: Uma Análise de Implementações em C, Java e Python

Análise teórica do Funcionamento e Complexidade

A análise da eficiência dos algoritmos é central para a Ciência da Computação, focando na taxa de crescimento do tempo de execução em função do tamanho da entrada (n). Essa avaliação é feita por meio da notação assintótica (O, Ω, Θ).

O processo de análise assume que cada operação elementar (como atribuição, comparação, ou operação aritmética) leva um tempo constante fixo.

A Busca Linear

A Busca Linear, ou LINEAR-SEARCH, é um algoritmo fundamental cuja tarefa é determinar se um valor particular (x) está presente em um arranjo (A).

Implementações da Busca Linear

Tabela 1. Implementação em pseudocódigo de uma Busca Linear

Linha	Pseudocódigo	Custo	Execução
1	Ajustamos resposta para NOT-FOUND	c1	1
2	Para i = 1 até n.	c2	n + 1
2a	Se A[i] = x, então ajuste resposta para o valor de i	c2a	n
3	Retorne o valor de resposta como saída.	c3	1

Implementações da Busca Linear

O pseudocódigo demonstra que o laço principal é executado um número fixo de vezes, n, independentemente dos dados de entrada.

Calculando…

$$T(n) = c_1 \cdot 1 + c_2 \cdot (n+1) + c_{2a} \cdot n + c_3 \cdot 1$$

$$T(n) = (c_2 + c_{2a}) \cdot n\,(c_1 + c_2 + c_3)$$

Implementações da Busca Linear

Tabela 2. Comparação das complexidades de tempos da Busca Linear

Caso	Complexidade de tempo
Pior caso	$\Theta(n)$
Melhor caso	$\Theta(n)$
Caso médio	$\Theta(n)$

Implementações da Busca Linear (Melhorada)

A Busca Linear Melhorada otimiza o algoritmo padrão ao retornar imediatamente o índice assim que o elemento procurado (x) é encontrado. Essa alteração torna o número de iterações do laço dependente da posição de x na entrada.

Tabela 3. Implementação em pseudocódigo de uma Busca Linear Melhorada

Linha	Pseudocódigo	Custo	Execução
1	Para i = 1 até n.	c1	n + 1
1a	Se A[i] = x, então retorne o valor de i como saída.	c1a	n
2	Retorne NOT-FOUND como saída.	c2	1

Implementações da Busca Linear (Melhorada)

O pior caso ocorre quando o algoritmo executa o tempo máximo possível. Isso acontece se o valor x não estiver presente no arranjo ou se for o último elemento (A[n]), forçando o laço a completar n iterações

$$T(n) = c_1 \cdot (n+1) + c_{1a} \cdot n + c_2 \cdot 1$$

$$T(n) = c_1 \cdot n + c_1 + c_{1a} \cdot n + c_2$$

$$T(n) = (c_1 + c_{1a}) \cdot n + (c_1 + c_2)$$

Implementações da Busca Linear (Melhorada)

Tabela 4. Comparação da complexidade de tempo entre três situações

Caso	Complexidade de tempo
Pior caso	$\Theta(n)$
Melhor caso	$\Theta(1)$
Caso médio	$\Theta(n)$ (Proporcional a n/2, que é $\Theta(n)$

Implementações da Busca Linear (Sentinela)

Esta variação visa reduzir o custo por iteração do laço. O algoritmo insere o valor procurado (x) na última posição do arranjo (A[n]), agindo como um sentinela. Isso permite eliminar a verificação da condição de parada do arranjo ($i \leq n$) dentro do laço principal, pois a presença do sentinela garante que o laço sempre terminará.

Implementações da Busca Linear (Sentinela)

Tabela 5. Implementação em pseudocódigo de uma Busca Linear com Sentinela

Linha	Pseudocódigo	Custo	Execução
1	Salve A[n] em último e então ponha x em A[n].	c1	1
2	Iguale i a 1.	c2	1
3	Enquanto A[i] = x, faça o seguinte:	c3	n
3a	Incremente i.	c3a	n - 1
4	Restaure A[n] de último.	c4	1
5	Se i < n ou A[n] = x, retorne o valor de i como saída.	c5	1
6	Caso contrário, retorne NOT-FOUND como saída.	c6	1

Implementações da Busca Linear (Sentinela)

O pior caso (quando x não está no arranjo original) ainda requer n iterações até que o sentinela em A[n] seja encontrado. O tempo de execução, T(n), é uma função linear de n:

$$T(n) = c_1 \cdot 1 + c_2 \cdot 1 + c_3 \cdot n + c_{3a} \cdot (n-1) + c_4 \cdot 1 + c_5 \cdot 1 + c_6 \cdot 1$$

$$T(n) = (c_3 + c_{3a}) \cdot n \cdot (c_1 + c_2 - c_{3a} + c_4 + c_5 + c_6)$$

$$T(n) = (c_3 + c_{3a}) \cdot n$$

Embora esta versão possua um fator constante menor multiplicando n (tornando-a potencialmente mais rápida na prática), a ordem de crescimento assintótica do pior caso permanece $\Theta(n)$.

Implementações da Busca Linear (Sentinela)

O pior caso (quando x não está no arranjo original) ainda requer n iterações até que o sentinela em A[n] seja encontrado. O tempo de execução, T(n), é uma função linear de n:

$$T(n) = c_1 \cdot 1 + c_2 \cdot 1 + c_3 \cdot n + c_{3a} \cdot (n-1) + c_4 \cdot 1 + c_5 \cdot 1 + c_6 \cdot 1$$

$$T(n) = (c_3 + c_{3a}) \cdot n \cdot (c_1 + c_2 - c_{3a} + c_4 + c_5 + c_6)$$

$$T(n) = (c_3 + c_{3a}) \cdot n$$

Embora esta versão possua um fator constante menor multiplicando n (tornando-a potencialmente mais rápida na prática), a ordem de crescimento assintótica do pior caso permanece $\Theta(n)$.

Resultados e Discussão

Os experimentos foram realizados em um computador com sistema operacional macOS, equipado com um processador Apple Silicon M1 e 8GB de memória RAM. Para garantir consistência e confiabilidade nas medições, cada algoritmo foi executado 50 vezes para cada tamanho de vetor, variando de 10.000 a 100.000 elementos.

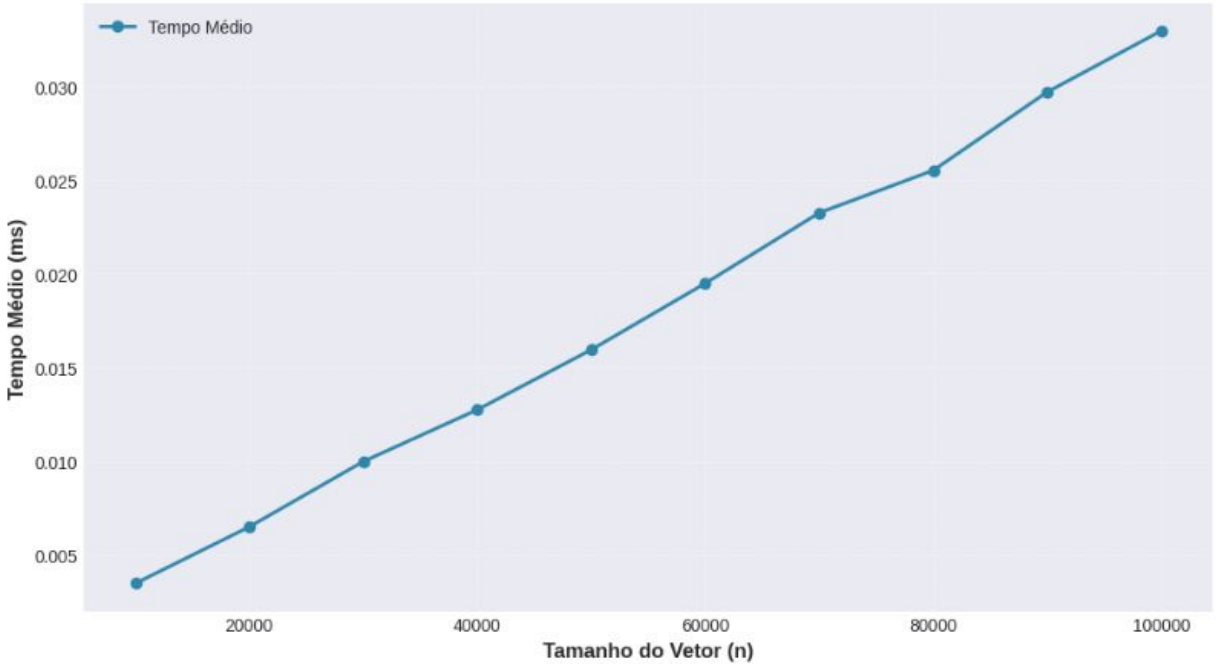
Resultados e Discussão

Os experimentos foram realizados em um computador com sistema operacional macOS, equipado com um processador **Apple Silicon M1 e 8GB de memória RAM.** Para garantir consistência e confiabilidade nas medições, cada algoritmo foi executado **50 vezes** para cada tamanho de vetor, variando de **10.000 a 100.000 elementos.**

Apple Clang 17.0.0 para o código em C, **JDK 25** para o programa em Java e **Python 3.13.3** para a versão em Python.

Resultados e Discussão

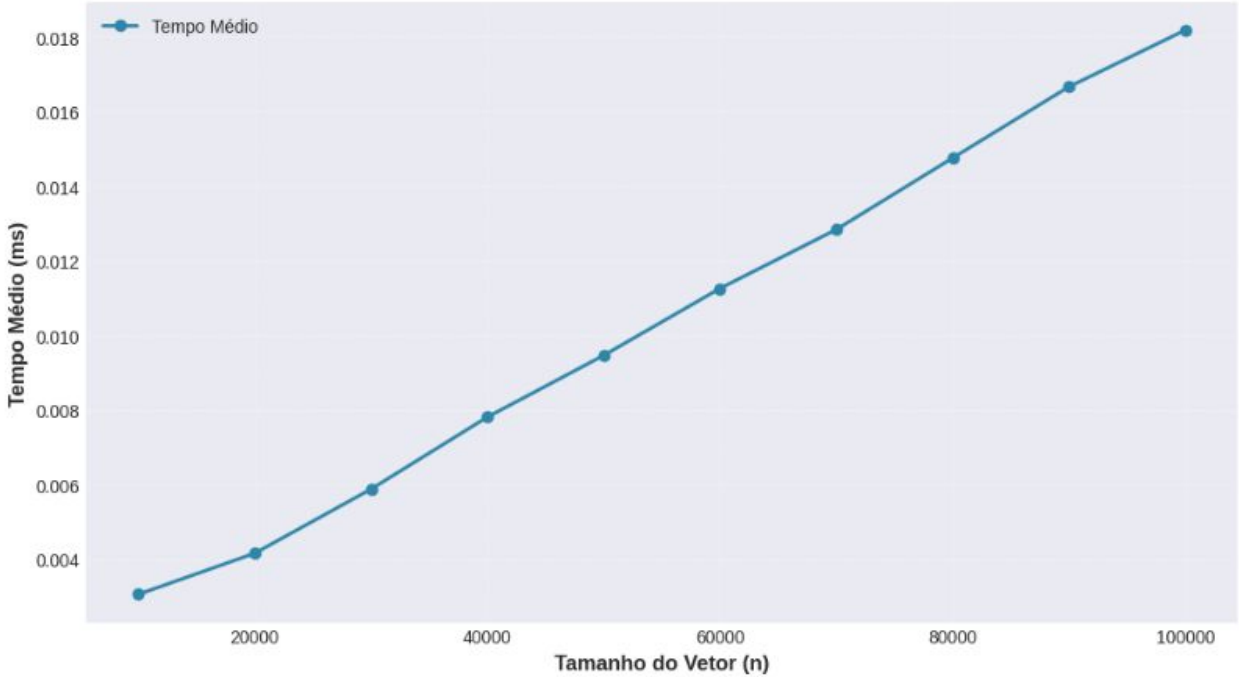
Figura 1. Desempenho do Algoritmo de Busca Linear em C



Fonte: O autor

Resultados e Discussão

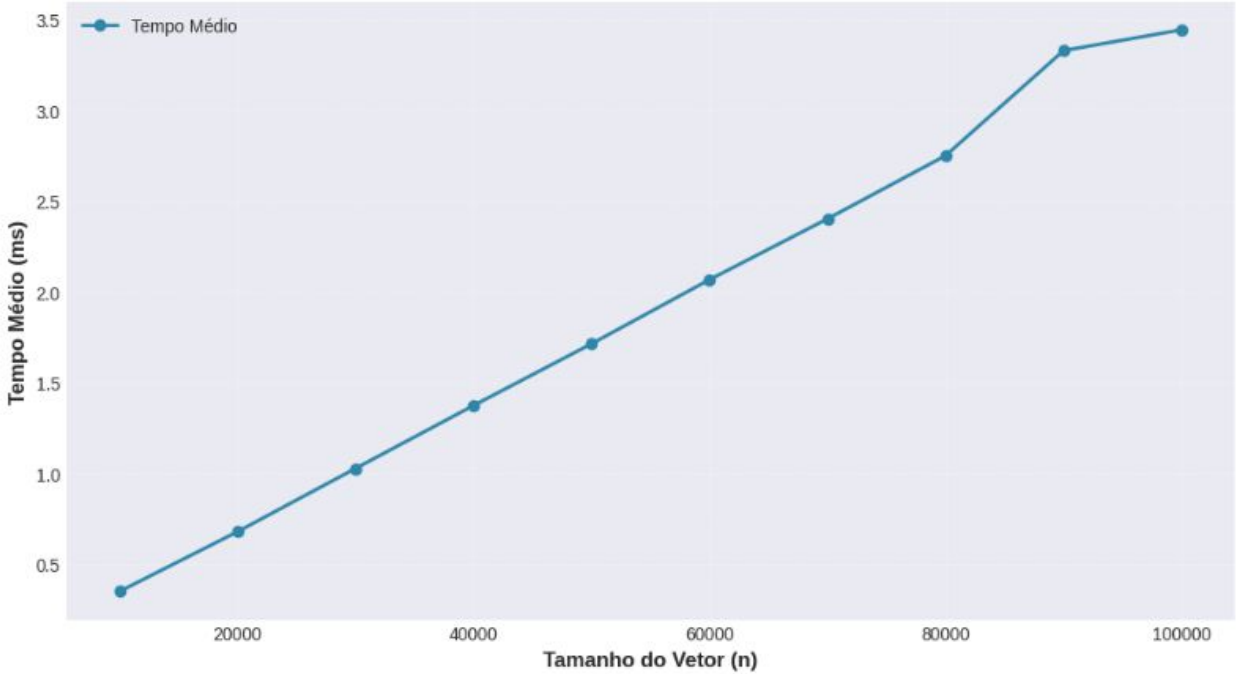
Figura 2. Desempenho do Algoritmo de Busca Linear em Java



Fonte: O autor

Resultados e Discussão

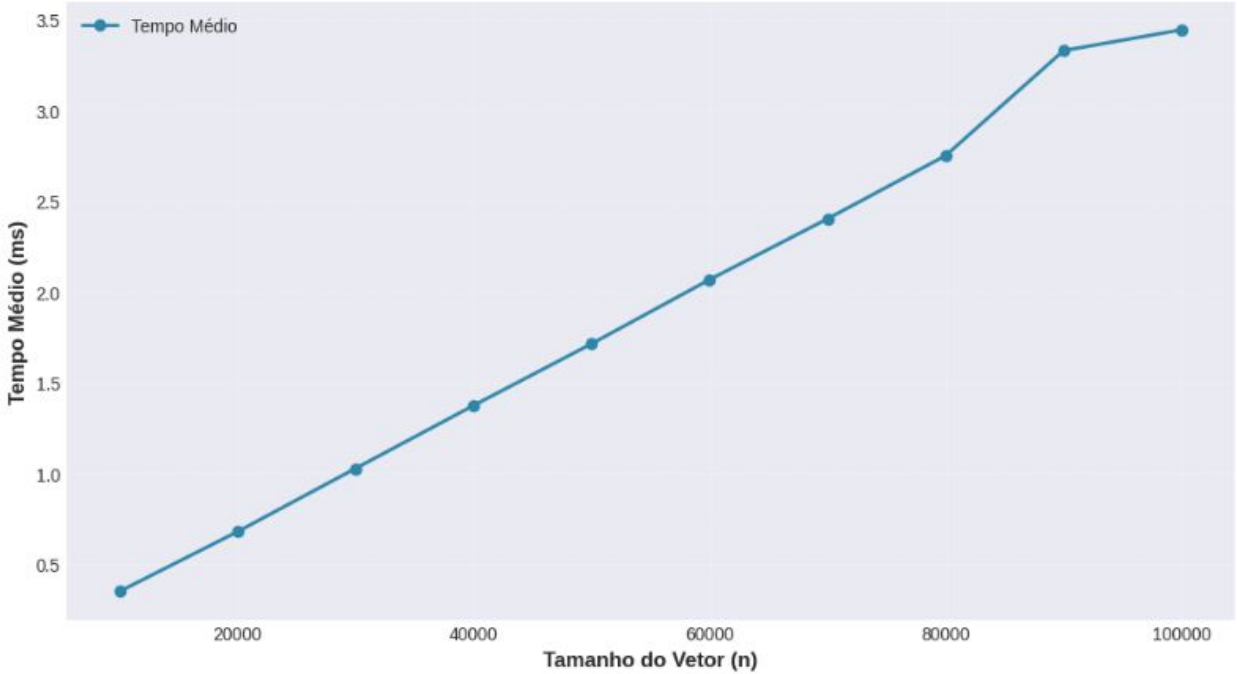
Figura 3. Desempenho do Algoritmo de Busca Linear em Python



Fonte: O autor

Resultados e Discussão

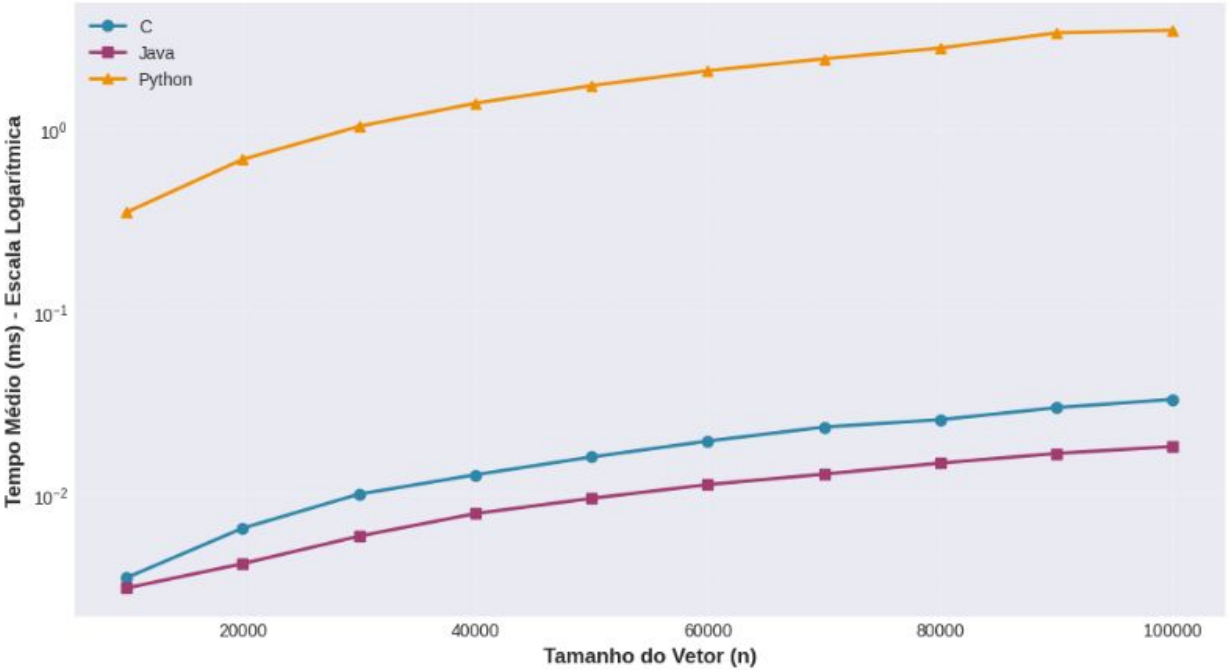
Figura 3. Desempenho do Algoritmo de Busca Linear em Python



Fonte: O autor

Resultados e Discussão

Figura 4. Comparativo de Desempenho - Busca Linear (C vs Java vs Python)



Fonte: O autor

Resultados e Discussão

A análise assintótica do funcionamento da Busca Linear (Padrão, Melhorada, com Sentinela e Recursiva) confirmou que, para o pior caso e o caso médio, o tempo de execução é caracterizado como $\Theta(n)$, indicando um crescimento linear em relação ao tamanho da entrada n. **O tempo de execução da Busca Linear é uma função linear da forma $c \cdot n + d$.**

Referências Bibliográficas

CORMEN, Thomas H. (2014), Desmistificando algoritmos. Tradução: Arlete Simille Marques. 1. ed. Rio de Janeiro: Elsevier

CORMEN, Thomas H.; LEISERSON, Charles E.; RIVEST, Ronald L.; STEIN, Clifford. (2022), Introduction to Algorithms. Fourth Edition. Cambridge, Massachusetts; London, England: The MIT Press