Relatório de Análise de Algoritmos: Inversão, Busca Sequencial e Busca Binária

Eduardo Malafronte Alves de Souza (nusp: 16862798)
Humberto Henrique de Amorim (nusp: 16814612)
Lucas Vinicius da Costa (nusp: 16885265)

22 de setembro de 2025

Sumário

1 Introdução

O objetivo geral deste trabalho é implementar, analisar e comparar a eficiência de diferentes algoritmos: a inversão vetorial, a busca sequencial e a busca binária, tanto em sua forma iterativa quanto recursiva. A análise foi conduzida de forma experimental, através da medição do tempo de execução, e teórica, pela contagem de operações de comparação e atribuição no pior caso. Para os testes, foram utilizadas entradas de 10, 100, 1.000 e 5.000 elementos, e o tempo médio foi calculado a partir de, no mínimo, 100 execuções para garantir a precisão dos resultados.

2 Análise dos Algoritmos

Nesta seção, apresentamos os resultados obtidos para cada algoritmo, incluindo o gráfico de tempo de execução e a análise teórica de complexidade no pior caso.

2.1 Inversão de Vetor

O gráfico de análise de desempenho do algoritmo de inversão vetorial explicita o caráter linear do crescimento do tempo de execução conforme o aumento do tamanho da entrada, comportamento esse esperado para um algoritmo que precisa percorrer o vetor.

Figura 1: Tempo de execução médio para o algoritmo de Inversão de Vetor.

Análise de Complexidade (Pior Caso): O algoritmo de inversão percorre o vetor até a sua metade, realizando uma operação de troca a cada iteração. Uma troca envolve 3 atribuições. O laço "for"executa $N/2$ vezes. A contagem de operações é aproximadamente:

$$T(N) = 5\lfloor\frac{N}{2}\rfloor + 2$$

Isso resulta em uma complexidade de tempo de $O(N)$.

Resultados Experimentais - Operações Médias:

Tamanho da Entrada (N)	**Operações Médias**
10	27,00
100	252,00
1.000	2.502,00
5.000	12.502,00

Tabela 1: Número de operações médias para o algoritmo de Inversão de Vetor

A complexidade teórica de $O(N)$ justifica o comportamento linear observado na análise gráfica, confirmando que o tempo de execução do algoritmo escala proporcionalmente com o tamanho da entrada.

2.2 Busca Sequencial

O gráfico de desempenho para o algoritmo de busca sequencial também demonstra um crescimento linear do tempo de execução. Assim como no algoritmo de inversão, este comportamento é esperado, pois, no pior caso, a busca sequencial precisa percorrer todos os N elementos do vetor.

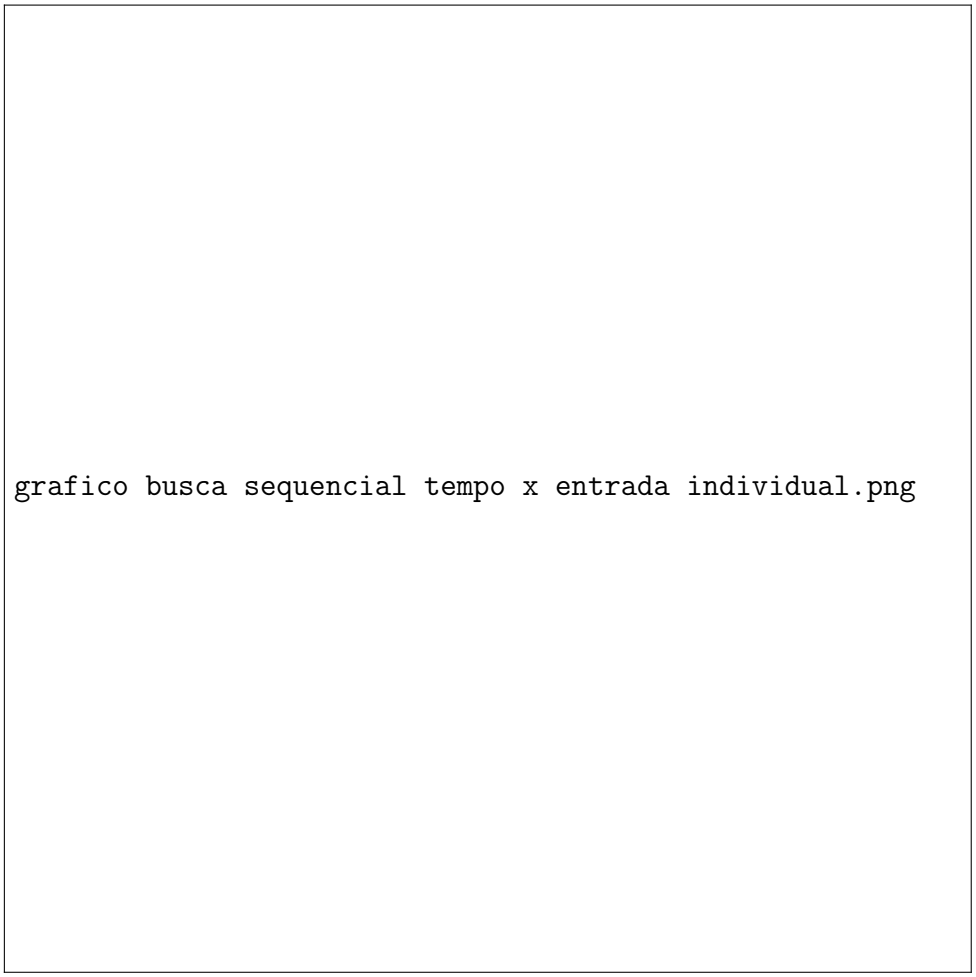

Figura 2: Tempo de execução médio (pior caso) para a Busca Sequencial.

Análise de Complexidade (Pior Caso): O pior caso ocorre quando o elemento procurado não está no vetor ou é o último. Nesse cenário, o algoritmo realiza uma comparação para cada um dos N elementos do vetor.

$$T(N) = 3N + 2$$

Onde temos N comparações no laço, N incrementos do contador, N comparações com o elemento procurado, uma comparação inicial e uma operação de retorno. A complexidade de tempo é, portanto, $O(N)$.

Resultados Experimentais - Operações Médias:

Tamanho da Entrada (N)	**Operações Médias**
10	32,00
100	302,00
1.000	3.002,00
5.000	15.002,00

Tabela 2: Número de operações médias para o algoritmo de Busca Sequencial

A complexidade teórica de $O(N)$ justifica o comportamento linear observado na análise gráfica, confirmando que o tempo de execução do algoritmo escala proporcionalmente com o tamanho da entrada.

2.3 Buscas Binárias (Iterativa e Recursiva)

Os algoritmos de busca binária apresentaram desempenho superior aos algoritmos lineares, confirmando sua característica logarítmica fundamental.

Figura 3: Tempo de execução médio (pior caso) para a Busca Binária Iterativa.

Figura 4: Tempo de execução médio (pior caso) para a Busca Binária Recursiva.

Análise de Complexidade (Pior Caso):

Busca Binária Iterativa: A cada iteração, o algoritmo realiza comparações para determinar em qual metade continuar a busca, além de operações de atualização dos índices:

$$T(N) = 5\lfloor \log_2(N) \rfloor + 3$$

Busca Binária Recursiva: A implementação recursiva adiciona o overhead das chamadas de função:

$$T(N) = 4\lfloor \log_2(N) \rfloor + 1$$

Resultados Experimentais - Operações Médias:

Busca Binária Iterativa:

Tamanho da Entrada (N)	Operações Médias
10	23,00
100	38,00
1.000	53,00
5.000	68,00

Tabela 3: Número de operações médias para a Busca Binária Iterativa

Busca Binária Recursiva:

Tamanho da Entrada (N)	Operações Médias
10	21,00
100	36,00
1.000	51,00
5.000	66,00

Tabela 4: Número de operações médias para a Busca Binária Recursiva

Ambas as versões mantêm complexidade de tempo $O(\log N)$. Os dados experimentais confirmam o comportamento logarítmico esperado, com a versão iterativa apresentando desempenho consistentemente similar à recursiva em termos de número de operações.

3 Conclusão

A análise experimental dos quatro algoritmos revelou duas classes distintas de complexidade e desempenho, $O(\log N)$ e $O(N)$. Para visualizar essa diferença de forma direta, o gráfico de desempenho comparativo entre todos os algortimos apresenta as curvas de tempo de execução deles em um único eixo.

Figura 5: Comparativo de desempenho entre os algoritmos analisados (Pior caso).

O gráfico indica claramente que as curvas dos algoritmos de Inversão de Vetor e Busca Sequencial crescem de forma linear e acentuada, confirmando sua complexidade $O(N)$. Em contraste, as curvas das Buscas Binárias (Iterativa e Recursiva) permanecem praticamente planas na base do gráfico, mal registrando aumento de tempo mesmo com 5.000 elementos. Essa representação visual demonstra a imensa escalabilidade e eficiência dos algoritmos de complexidade logarítmica ($O(\log N)$) em comparação com os lineares ($O(N)$).

Para entradas pequenas, a diferença de tempo pode ser irrelevante, mas para sistemas que manipulam grandes quantidades de dados, a escolha de um algoritmo $O(\log N)$ em vez de um $O(N)$ é de suma importância para a performance. Portanto, é possível concluir

que os resultados experimentais obtidos foram consistentes com a teoria de análise de algoritmos, demonstrando a importância do estudo da complexidade para o desenvolvimento de soluções computacionais eficientes.

4 Contribuição dos Membros da Equipe

A seguir, a descrição das principais contribuições de cada membro da equipe:

- **Eduardo Malafronte Alves de Souza:**
  - Criação de todos os gráficos para a visualização e análise do tempo de execução dos algoritmos.
  - Redação, estruturação e formatação do relatório, consolidando os resultados e as análises da equipe.
- **Humberto Henrique de Amorim:**
  - Coleta e tabulação dos dados de tempo de execução.
  - Análise teórica da contagem de operações dos algoritmos.
- **Lucas Vinicius da Costa:**
  - Responsável pelo desenvolvimento da solução em linguagem C, implementando os quatro algoritmos solicitados.