**ESCOLA POLITÉCNICA**

**ENGENHARIA DE SOFTWARE**

**PEDRO HENRIQUE SANTOS DE ARAUJO RA 22900732**

**Estudo Comparativo de Algoritmos de Busca**

**Atividade de Recuperação - Avaliação P2**

ATIVIDADE DE RECUPERAÇÃO, apresentado no componente curricular Fundamentos de Teoria da Computação e Análise de Algoritmos, do curso de Engenharia de Software, da Escola Politécnica da Pontifícia Universidade Católica de Campinas.

Orientador: Jose Guilherme Picolo

**CAMPINAS**

**2025**

[**INTRODUÇÃO** 2](#_Toc200451318)

[**OBJETIVO DO ESTUDO** 2](#_Toc200451319)

[**FUNCIONAMENTO DOS ALGORITMOS** 2](#_Toc200451320)

[**COMPLEXIDADE DE TEMPO ESPERADO PELO ALGORITMO** 3](#_Toc200451321)

[**METODOLOGIA** 3](#_Toc200451322)

[**RESULTADOS** 4](#_Toc200451323)

[**CONCLUSÃO** 5](#_Toc200451324)

[**BIBLIOGRAFIA** 5](#_Toc200451325)

# **INTRODUÇÃO**

O Relatório demonstra uma análise comparativa entre os algoritmos fundamentais de busca. A busca sequencial e a busca binária. O estudo visa a demonstração do comportamento dos algoritmos através de tamanhos de entradas controlados, bem como a validação se o algoritmo condiz com as expectativas teóricas estabelecidas pela análise assintótica.

# **OBJETIVO DO ESTUDO**

Comparar os algoritmos de busca sequencial e binária considerando:

* Tempo de execução
* Número de posições visitadas
* Casos de teste (médio e pior)
* Diferentes tamanhos de entrada (1.000, 10.000 e variável)

# **FUNCIONAMENTO DOS ALGORITMOS**

**Busca Sequencial**

* Percorre sequencialmente cada elemento do vetor até o fim da lista. Simples de implementar, porém muito ineficiente em grandes porções de dados

**Busca Binária**

Exige primeiramente que a lista esteja ordenada. A cada iteração reduz pela metade o intervalo de busca, comparando o valor alvo com o elemento central da lista.

# **COMPLEXIDADE DE TEMPO ESPERADO PELO ALGORITMO**

|  |  |  |  |
| --- | --- | --- | --- |
| Algoritmo | Melhor Caso | Caso Médio | Pior Caso |
| Sequencial | O(1) | O(n) | O(n) |
| Binária | O(1) | O(log n) | O(log n) |

# **METODOLOGIA**

**Hardware utilizado:**

* Processador : i7 6500u
* Memória RAM: 8GB DDR4
* Placa de Vídeo : GTX880M
* Sistema Operacional : Windows 11 Pro

**Linguagem de Programação**

* Python 3.13
* Bibliotecas: time, Random, tkinter

**Repositório do Código**

* <https://github.com/PedroHenriqueAraujo18/Atividade-de-Recupera-o---Avalia-o-P2>

# **RESULTADOS**

* **Caso Médio – Alvo Presente**

|  |  |  |  |
| --- | --- | --- | --- |
| Tamanho | Linear Search | Visitas | Tempo(s) |
| 1000 | O(n) | 37 | 0.000034 |

|  |  |  |  |
| --- | --- | --- | --- |
| Tamanho | Binary Search | Visitas | Tempo(s) |
| 1000 | O(log n) | 10 | 0.000034 |

|  |  |  |  |
| --- | --- | --- | --- |
| Tamanho | Linear Search | Visitas | Tempo(s) |
| 10000 | O(n) | 2004 | 0.001437 |

|  |  |  |  |
| --- | --- | --- | --- |
| Tamanho | Binary Search | Visitas | Tempo(s) |
| 10000 | O(log n) | 13 | 0.000029 |

|  |  |  |  |
| --- | --- | --- | --- |
| Tamanho | Linear Search | Visitas | Tempo(s) |
| 50000 | O(n) | 8610 | 0.005324 |

|  |  |  |  |
| --- | --- | --- | --- |
| Tamanho | Binary Search | Visitas | Tempo(s) |
| 50000 | O(log n) | 16 | 0.000053 |

* **Pior Caso – Alvo Ausente**

|  |  |  |  |
| --- | --- | --- | --- |
| Tamanho | Linear Search | Visitas | Tempo(s) |
| 1000 | O(n) | 1000 | 0.001600 |

|  |  |  |  |
| --- | --- | --- | --- |
| Tamanho | Binary Search | Visitas | Tempo(s) |
| 1000 | O(log n) | 9 | 0.000044 |

|  |  |  |  |
| --- | --- | --- | --- |
| Tamanho | Linear Search | Visitas | Tempo(s) |
| 10000 | O(n) | 10000 | 0.003645 |

|  |  |  |  |
| --- | --- | --- | --- |
| Tamanho | Binary Search | Visitas | Tempo(s) |
| 10000 | O(log n) | 13 | 0.000014 |

|  |  |  |  |
| --- | --- | --- | --- |
| Tamanho | Linear Search | Visitas | Tempo(s) |
| 50000 | O(n) | 50000 | 0.029019 |

|  |  |  |  |
| --- | --- | --- | --- |
| Tamanho | Binary Search | Visitas | Tempo(s) |
| 50000 | O(log n) | 15 | 0.000019 |

# **CONCLUSÃO**

**Análise Crítica:**

* A busca sequencial é simples e de fácil implementação, mas ineficiente para grandes listas. A busca binária, apesar de exigir lista ordenada, entrega resultados muito superiores em desempenho.

**Vantagens e Limitações:**

* **Busca Sequencial**: flexível, simples de implementar, mas muito lenta em grandes bases de dados
* **Busca Binária**: rápida, mas limitada às listas ordenadas.

**Reflexão Final:**

* A escolha de um algoritmo deve sempre considerar o contexto e a eficiência desejada. A análise de complexidade é essencial para garantir soluções escaláveis e otimizadas.

# **BIBLIOGRAFIA**

* Cormen, T. H. et al. *Introduction to Algorithms*. MIT Press.
* <https://www.geeksforgeeks.org/binary-search/>
* <https://docs.python.org/3/library/tkinter.html>