

# Comparação dos Algoritmos de Soma Máxima de Subsequência

Pablo Augusto Matos da Silva  
Matrícula 2022015139

23 de março de 2025

## Introdução

Este relatório apresenta a análise de desempenho de três algoritmos para encontrar a soma máxima de uma subsequência contígua em um vetor de inteiros. Os algoritmos avaliados foram:

1. **Força Bruta  $O(n^2)$**  - Verifica todas as possíveis subsequências.
2. **Divisão e Conquista  $O(n \log n)$**  - Divide o problema recursivamente.
3. **Kadane  $O(n)$**  - Resolve de forma iterativa com uma abordagem eficiente.

Os testes foram realizados com diferentes tamanhos de entrada, e foram coletados os seguintes dados:

- **Resultado:** Soma máxima encontrada.
- **Tempo de execução:** Tempo medido em segundos.
- **Quantidade de operações realizadas:** Contagem de operações básicas dentro de cada algoritmo.

## Resultados

Arquivo: 1M.dat | Tamanho: 1024

- **Força Bruta  $O(n^2)$**  -> Resultado: 18520 | Tempo: 0.002162 s | Operações: 524894
- **Divisão e Conquista  $O(n \log n)$**  -> Resultado: 18520 | Tempo: 0.000122 s | Operações: 14500
- **Kadane  $O(n)$**  -> Resultado: 18520 | Tempo: 0.000007 s | Operações: 1132

**Arquivo: 4M.dat | Tamanho: 4096**

- **Força Bruta  $O(n^2)$**  -> Resultado: 183316 | Tempo: 0.017244 s | Operações: 8390830
- **Divisão e Conquista  $O(n \log n)$**  -> Resultado: 183316 | Tempo: 0.000242 s | Operações: 67055
- **Kadane  $O(n)$**  -> Resultado: 183316 | Tempo: 0.000011 s | Operações: 4312

**Arquivo: 8M.dat | Tamanho: 8192**

- **Força Bruta  $O(n^2)$**  -> Resultado: 243534 | Tempo: 0.054671 s | Operações: 33558750
- **Divisão e Conquista  $O(n \log n)$**  -> Resultado: 243534 | Tempo: 0.000445 s | Operações: 142588
- **Kadane  $O(n)$**  -> Resultado: 243534 | Tempo: 0.000019 s | Operações: 8390

## Detalhamento dos Testes

Para realização dos experimentos foi utilizado com computador com as seguintes configurações:

### Hardware

- **Processador:** Intel® Core™ - i7-12700H Alder Lake - 2.3GHz (TurboMax 4.7GHz) - 24MB cache
- **Memória:** 16GB [2x 8GB - Dual Channel] Memória DDR4 (3200 MHZ)

### Software

- **Sistema Operacional:** Pop! OS 22.04 LTS 64 bits
- **Linguagem:** C17 (201710)
- **Compilador:** gcc (Ubuntu 11.4.0-1ubuntu1 22.04) 11.4.0

Para mais detalhes sobre o código acesse o seguinte repositório online no Github:  
[https://github.com/pabloaugmatrix/programacao\\_paralela/tree/main/atividade02](https://github.com/pabloaugmatrix/programacao_paralela/tree/main/atividade02)

## Análise e Conclusões

Os resultados confirmam a previsão teórica sobre a complexidade dos algoritmos:

- **Força Bruta:** Apresentou crescimento quadrático em tempo e operações, podendo se tornar inviável para grandes tamanhos de entrada.
- **Divisão e Conquista:** Escala melhor que a força bruta, mas ainda tem um crescimento notável no número de operações.
- **Kadane:** Mostrou desempenho significativamente superior, com tempo e operações crescendo de forma linear, confirmando sua eficiência.

Dessa forma, para aplicações que demandam alta performance, o **algoritmo de Kadane** é claramente a melhor escolha. No entanto, o algoritmo de **Divisão e Conquista** pode ser útil em situações que requerem abordagem recursiva.