

Universidade Federal de Ouro Preto  
Lecture Notes  
Introdução à análise de algoritmos

Prof. Rodrigo Silva

August 28, 2023

## Leitura Recomendada

- Capítulo 2 - *Introduction to the Design and Analysis of Algorithms (3rd Edition)* - Anany Levitin

## 1 Definições

**Definition 1.1 (Operação básica)** *Operação que mais contribui para o tempo total de execução.*

### 1.1 Melhor caso, pior caso e caso médio

Claro, vou definir cada um desses conceitos no contexto de análise de algoritmos:

**Definition 1.2 (Melhor Caso (Best Case))** *O melhor caso de um algoritmo é aquele em que ele alcança o menor consumo de recursos (tempo ou espaço) entre todas as entradas possíveis. Em outras palavras, é a situação em que o algoritmo se comporta da forma mais eficiente. No entanto, o melhor caso nem sempre é o cenário mais relevante, já que as entradas da vida real podem não se assemelhar ao melhor caso.*

**Definition 1.3 (Pior Caso (Worst Case))** *O pior caso de um algoritmo é aquele em que ele requer o maior consumo de recursos entre todas as entradas possíveis. Isso dá uma ideia do desempenho máximo que o algoritmo pode atingir. O pior caso é frequentemente utilizado para garantir que o algoritmo não terá um desempenho inaceitavelmente ruim, independentemente das entradas.*

**Definition 1.4 (Caso Médio (Average Case))** *O caso médio de um algoritmo é a média ponderada dos consumos de recursos (tempo ou espaço) em todas as possíveis entradas. Isso proporciona uma visão mais realista do desempenho do algoritmo, já que as entradas da vida real tendem a ser uma mistura de diferentes casos. A análise do caso médio pode ser complexa, pois requer o conhecimento das probabilidades de diferentes entradas e como o algoritmo se comporta em cada uma.*

**Definition 1.5 (Eficiência Amortizada (Amortized Efficiency))** *A eficiência amortizada é uma medida de desempenho que leva em consideração o custo total médio de uma sequência de operações ao longo do tempo. Isso é particularmente útil quando um algoritmo pode ter operações caras, mas, em média, elas são balanceadas por operações mais baratas. Um exemplo clássico é a análise de estruturas de dados dinâmicas, como tabelas de hash ou filas, onde algumas operações podem ser mais custosas em termos de tempo, mas, ao longo do tempo, o custo médio por operação é razoável.*

---

**Algorithm 1** Busca Sequencial

---

1: **function** BUSCASEQUENCIAL(*lista*, *alvo*)

2:   **for**  $i \leftarrow 1$  to *tamanho(lista)* **do**

3:     **if** *lista*[*i*] = *alvo* **then**

4:       **return** *i*

▷ Elemento encontrado na posição *i*

5:     **end if**

6:   **end for**

7:   **return**  $-1$

▷ Elemento não encontrado na lista

8: **end function**

---