

1. Resumo – Capítulo 1 do Livro de Algoritmos

1.1 Crescimento da Função (Big O)

Os gráficos representam como o tempo de execução cresce conforme o número de operações. Exemplos de curvas:

$O(\log n)$ – crescimento logarítmico

$O(n)$ – crescimento linear

$O(n * \log n)$ – crescimento linear

$O(n^2)$ – crescimento quadrático

$O(n!)$ – crescimento fatorial (extremamente rápido)

O que o Big O representa

O Big O representa o tempo de execução em relação ao número de operações realizadas pelo algoritmo.

A complexidade de um algoritmo não é medida em segundos, mas sim pelo crescimento do número de operações.

Ou seja:

Quanto maior o tamanho da entrada, maior será o tempo de execução porém esse aumento depende do tipo de algoritmo.

A notação Big O considera o pior caso.

Para casos simples, $O(n)$ é o tempo que leva para verificar todos os itens da lista.

2. Exemplos comuns de tempos de execução (Big O)

$O(\log n)$ – tempo logarítmico (pesquisa binária)

$O(n)$ – tempo linear (pesquisa simples)

$O(n * \log n)$ – quicksort

$O(n^2)$ – ordenação por seleção

$O(n!)$ – caixeiro viajante (tempo fatorial)

3. Exemplo do livro – Pesquisa simples x Pesquisa binária

Tamanho da lista	Pesquisa Simples	Pesquisa Binária
100 elementos	100 ms	7 ms
10.000 elementos	10 segundos	14 ms
1 bilhão	10 dias	32 ms

Quanto maior a lista, mais rápido a pesquisa binária fica em relação à pesquisa simples.

A notação Big O leva isso em consideração:

Pesquisa simples → $O(n)$

Pesquisa binária → $O(\log n)$

4. Algoritmos

Algoritmo 1 – Descobrir quantos passos

$$O(16) = 16$$

(Simplesmente significa 16 operações.)

Algoritmo 2 – Pesquisando no papel

$$O(\log n)$$

$$\text{Exemplo: } \log_2 16 = 4$$

5. Tabela comparativa (tempo linear x logarítmico)

Pesquisa Simples Pesquisa Binária

100 itens 7 palpites

100 palpites

1 bilhão de itens 32 palpites

1 bilhão tentativas

$O(n) \rightarrow$ tempo linear

$O(\log n) \rightarrow$ tempo logarítmico

6. Estudo do Livro – Algoritmos (Capítulo 1)

2.2 Exercícios

$$1.1 - \log_2 128 = 7$$

$$1.2 - \log_2 256 = 8$$

Pesquisa simples

Em uma pesquisa simples, o número de termos da lista é igual ao número máximo de tentativas.

Esse é um tempo linear.

3.2 Pesquisa binária

Se a lista tem 100 itens, precisa de 7 tentativas (pois $\log_2 100 \approx 7$).

Se a lista tiver 1 bilhão, $\log_2(1.000.000.000) \approx 32$ — muito menos!

Esse é um tempo logarítmico, não linear.