

# Um resumo sobre complexidade de algoritmos

Anna Gabriele Marques de Oliveira - 20200266  
UFPel - Ciência da computação

April 4, 2022

## Contents

<b>1</b>	<b>O que é complexidade de algoritmos?</b>	<b>1</b>
<b>2</b>	<b>Utilização complexidade de algoritmos</b>	<b>1</b>
<b>3</b>	<b>A notação Big O</b>	<b>1</b>
3.1	$O(1)$ : . . . . .	1
3.2	$O(\log n)$ : . . . . .	1
3.3	$O(n)$ : . . . . .	1
3.4	$O(n^2)$ : . . . . .	2
3.5	$O(2^n)$ : . . . . .	2
3.6	$O(n!)$ : . . . . .	2
<b>4</b>	<b>Praticidade comercial</b>	<b>2</b>
<b>5</b>	<b>contratação de programadores</b>	<b>2</b>

## 1 O que é complexidade de algoritmos?

A complexidade de um algoritmo é uma medida para determinar a quantidade de tempo que um algoritmo leva para processar dados de acordo com o tamanho de sua entrada.

## 2 Utilização complexidade de algoritmos

A preocupação com a complexidade de algoritmos é fundamental para projetar algoritmos eficientes. Um algoritmo eficiente utiliza da melhor maneira possível a capacidade computacional e os recursos existentes.

## 3 A notação Big O

### 3.1 $O(1)$ :

Caracterizada nos programas que possuem “operações de custo fixo”. Onde tempo de execução de um programa é constante e não depende da entrada.

- Exemplos:
  - Uma função que apenas imprime um número, e o número é escolhido pelo programador na chamada da função.[1]
  - Uma função que determina se o número é ímpar ou par.[2]

### 3.2 $O(\log n)$ :

Caracterizada nos programas que utilizam da técnica de “dividir para conquistar”.

- Exemplos:
  - Para buscar um nome na lista telefônica não a percorremos por completo, buscamos apenas os nomes que possuem a primeira letra correspondente ao nome requerido.[1]
  - Encontrar um elemento em uma árvore binária perfeitamente balanceada.[2]

### 3.3 $O(n)$ :

Caracterizada nos programas percorrem um loop e que o resultado depende do tamanho da entrada.

- Exemplos:
  - Um algoritmo que recebe  $n$  números e imprime todos os números de 1 até  $n$ .
  - Um algoritmo sort para ordenar um vetor.

### 3.4 $O(n^2)$ :

Quando o algoritmo performa uma complexidade  $O(n)$  para cada entrada.

- Exemplos:
  - Um problema de multiplicação de matrizes quadradas. [2]

### 3.5 $O(2^n)$ :

Caracterizada pelas funções exponenciais, esses programas tem a sua taxa de crescimento dobrada a cada entrada.

- Exemplos:
  - O cálculo recursivo dos números de Fibonacci.
  - A torre de Hanói:



Figure 1: Torre de Hanói. [3]

### 3.6 $O(n!)$ :

- Exemplos:
  - O problema do caixeiro viajante pelo método da força bruta. Com  $N$  cidades, temos a permutação  $N!$ , tornando sua complexidade fatorial  $O(N!)$ . [4]

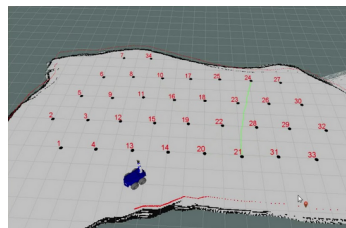


Figure 2: O problema do caixeiro viajante. [4]

## 4 Praticidade comercial

O big  $O$  é amplamente utilizado no âmbito comercial, uma vez que o mesmo prioriza a minimização das perdas de recursos. É utilizado, principalmente, em empresas que desejam otimizar o acesso em larga escala, onde a entrada se torna exponencialmente grande.

## 5 contratação de programadores

Neste vídeo eles utilizam complexidade de algoritmos para determinar o menor número de conexões que um avião fará entre um aeroporto central e os outros, sendo que não há conexão direta entre todos os aeroportos.

[Entrevista.](#)

## References

- [1] CódigoEscola. Notação big o e complexidade de algoritmos. <https://www.youtube.com/watch?v=X6LTVkymVM>, 2021.
- [2] I. Team. The big-o! time complexity with examples. [urlhttps://medium.com/swlh/the-big-o-time-complexity-with-examples-2b0a76a358b2](https://medium.com/swlh/the-big-o-time-complexity-with-examples-2b0a76a358b2), 2020.
- [3] Wikipedia. Torre de hanói. [https://pt.wikipedia.org/wiki/Torre\\_de\\_Hanói](https://pt.wikipedia.org/wiki/Torre_de_Hanói), 2022.
- [4] Wikipédia. Problema do caixeiro-viajante. [https://pt.wikipedia.org/wiki/Problema\\_do\\_caixeiro-viajante](https://pt.wikipedia.org/wiki/Problema_do_caixeiro-viajante), 2021.