

# Tópicos em Complexidade Computacional

## Apresentação da Disciplina e Definições Elementares

**Professor Murilo V. G. da Silva**

Departamento de Informática  
Universidade Federal do Paraná

23/05/2022

# Dificuldade de Problemas

Alguns problemas são mais difíceis que outros?

$\text{Strawberry} = 1$     $\text{Pear} = 2$     $\text{Flower} = 3$

$\text{Pear} + \text{Strawberry} + \text{Strawberry} = \square$

$\text{Flower} + \text{Flower} - \text{Strawberry} = \square$

$\text{Pear} + \text{Pear} + \text{Flower} = \square$

$\text{Flower} - \text{Pear} + \text{Strawberry} = \square$

$\text{Strawberry} + \text{Strawberry} + \text{Strawberry} = \square$

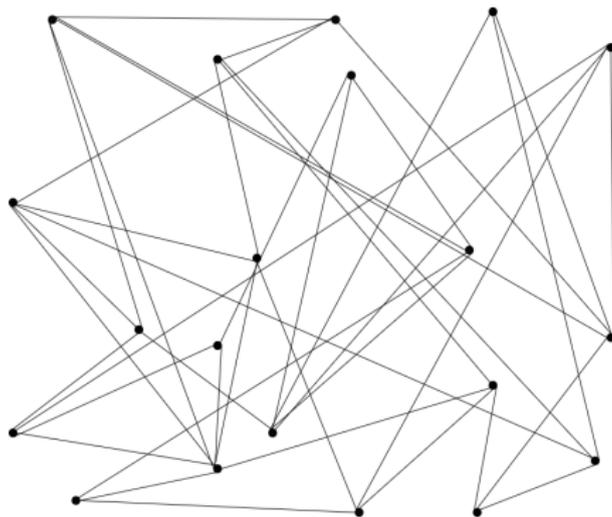
SOLVE EXAMPLES

1 2 3 4 5 6 7 8 9 10

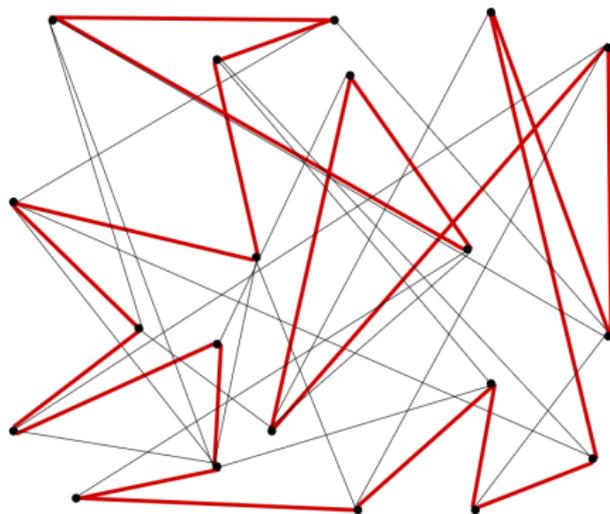
© dreamstime.com   © 177724136 © Fotografochka



Considere o problema do Grafo Hamiltoniano:

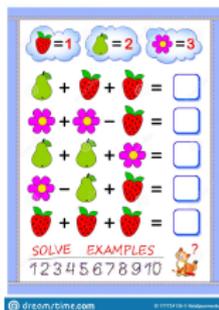


Considere o problema do Grafo Hamiltoniano:

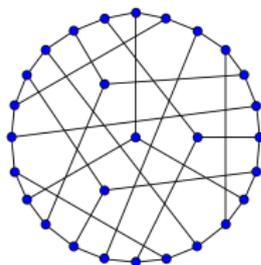


# Dificuldade de Problemas

Decisão polinomial vs Verificação polinomial vs Exponencial  
(até para verificar?)



Soma de inteiros



Grafo Hamiltoniano



Xadrez Generalizado

Exponencial é ruim? Explicação visual:

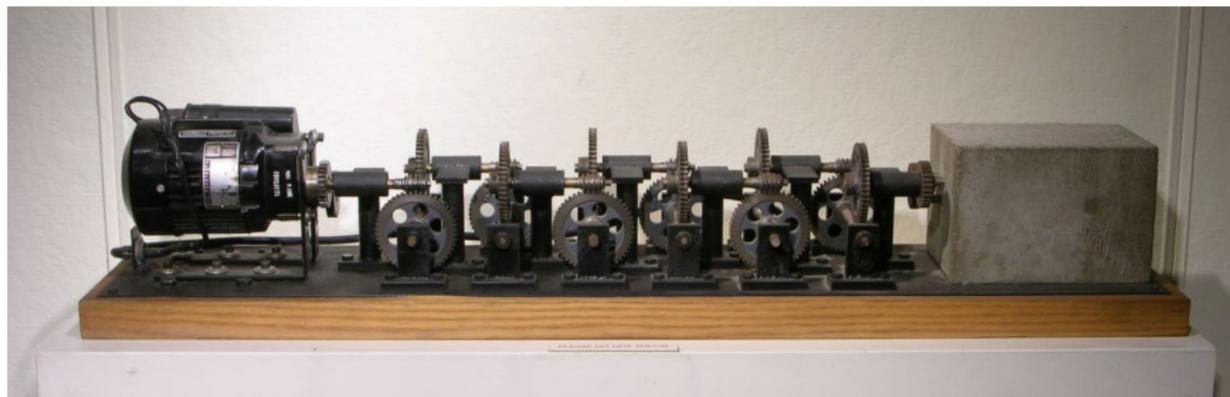
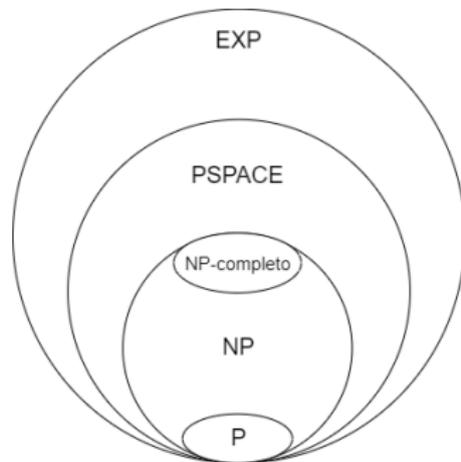


Figura: Barak & Arora, pág. 5

Algumas classes de complexidade estudadas em ITC



# Existem muito mais classes de complexidade!

Complexity Zoo

complexityzoo.net/Complexity\_Zoo

SEI Murilo V. G. da Silva gmail Calendar YouTube Drive CMSC 652 --- Com... 15-855: Graduate C... CS 151 Seminários Online

Create account Log in

Main page Discussion Read View source View history Search Complexity Zoo

**Complexity Zoo**

**Introduction**

Welcome to the **Complexity Zoo**. There are now 545 classes and counting!

Complexity classes by letter: [Symbols](#) - [A](#) - [B](#) - [C](#) - [D](#) - [E](#) - [F](#) - [G](#) - [H](#) - [I](#) - [J](#) - [K](#) - [L](#) - [M](#) - [N](#) - [O](#) - [P](#) - [Q](#) - [R](#) - [S](#) - [T](#) - [U](#) - [V](#) - [W](#) - [X](#) - [Y](#) - [Z](#)

*Lists of related classes:* [Communication Complexity](#) - [Hierarchies](#) - [Nonuniform](#)

**Zookeeper**  
[Scott Aaronson](#)

**Veterinarian**  
[Greg Kuperberg](#)

**Zoo Conservationist**  
[Oliver Habryka](#) on behalf of the [LessWrong](#) community

The Zoo first opened in 2002. It was made into a wiki in 2005, and hosted at the University of Waterloo from 2012 to 2020.

Errors? Omissions? Misattributions? Your favorite class not here? Then please contribute to the zoo as you see fit by [signing up](#) and clicking on the edit links. Please include references, or better yet links to papers if available.

To create a new class, click on the edit link of the class before or after the one that you want to add and copy the format of that class. (The classes are alphabetized by their tag names.) Then add the class to the table of contents and increment the total number of classes. After this, you can use the side edit links to edit the individual sections. For more on using the wiki language, see the [MediaWiki Help page](#).

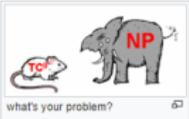
If you would like to contribute but feel unable to make the updates yourself, email the zookeeper at [scott@scottaaronson.com](mailto:scott@scottaaronson.com).

**See Also**

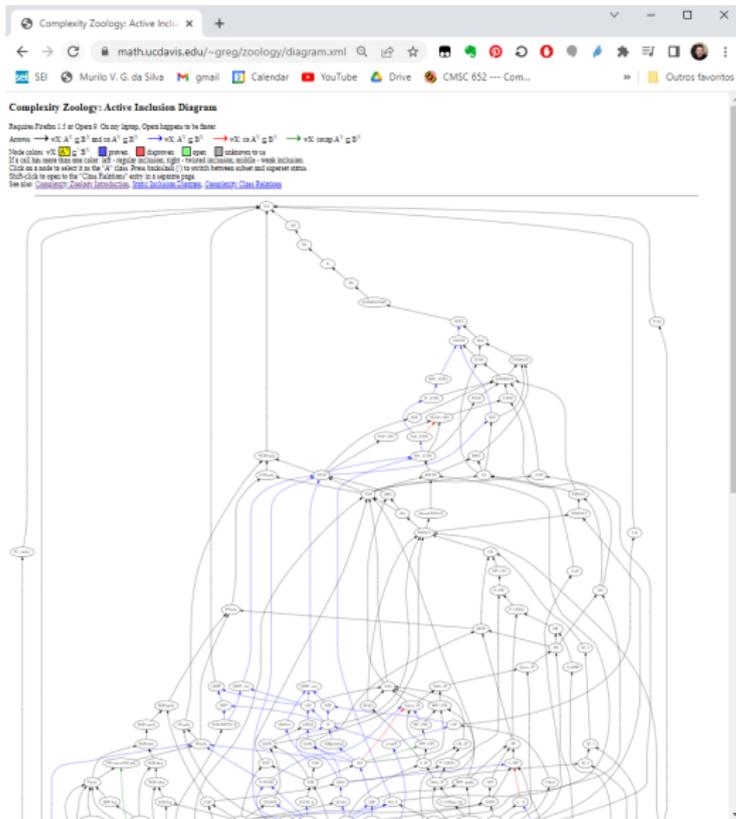
*Introductory Resources*

- [Introductory Essay](#): New visitors may want to stop here and see what the Zoo is all about.
- [Petting Zoo](#): A more gentle version of the Zoo with fewer classes, meant for new initiates in complexity. (If you're looking for where the Most Important Classes went, look in the Petting Zoo.)

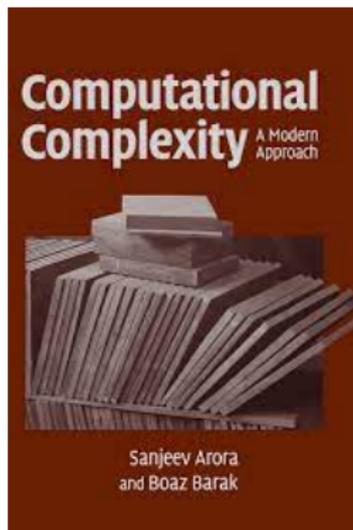
*Other Collections and Resources*



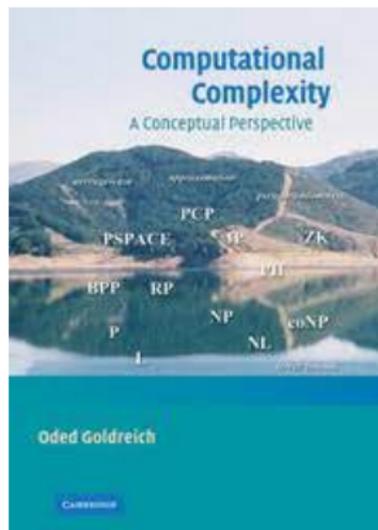
# Existem muito mais classes de complexidade!



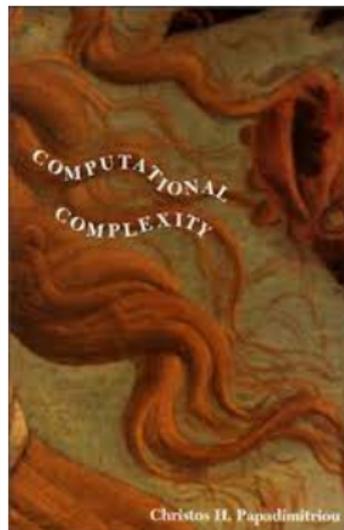
# Livros texto da disciplina



**Barak/Arora**



Goldreich



Papadimitriou

# Definições Elementares

Obs: Leiam a introdução e o capítulo 0 do livro de Barak/Arora para detalhes!

## Alfabetos

Seja  $S$  um alfabeto. (tipicamente  $S = \{0, 1\}$ )

- $S^n$  é o conjunto de strings de tamanho  $n$
- $S^*$  conjunto de todas as strings

## Strings

Seja  $x, y \in S^n$  strings (ou um vetores).

- $x_i$  é o  $i$ -ésimo bit de  $x$
- $x \odot y = \sum x_i y_i \pmod{2}$  (caso de reais/complexos:  $\langle x, y \rangle$ )

## Linguagens

A partir da função booleana  $f : \{0, 1\}^* \rightarrow \{0, 1\}$  definimos

- A linguagem  $L_f = \{x \mid f(x) = 1\}$
- Também nos referimos à  $L_f$  como um problema de decisão

Atenção: Notação usada em ITC

## Problema de Decisão (exemplos)

- $L_{\text{SAT}} = \{\perp\phi\perp \mid \phi \text{ é uma fórmula booleana em CNF satisfazível}\}$
- $L_{\text{IS}} = \{\perp(G, k)\perp \mid G \text{ é um grafo e } \exists S \subseteq V(G), |S| = k \text{ e } \forall u, v \in S, \{u, v\} \notin E(G)\}$

Atenção: Notação usada nesta disciplina (livro de Barak/Arora)

- $\text{INDSET} = \{\langle G, k \rangle : \exists S \subseteq V(G), |S| = k \text{ e } \forall u, v \in S, uv \notin E(G)\}$
- $\text{SAT} = \{\perp\phi\perp : \phi \text{ é uma fórmula booleana em CNF satisfazível}\}$ 
  - $\langle x, y \rangle$  notação para tupla (omitindo cod. em binário)
  - $uv$  representa aresta
  - Por convenção  $\perp A \perp$  sempre codifica objeto válido. Por **exemplo**,
    - se string que não é codificação válida, então à fórmula  $\overline{(x_1 \wedge \overline{x_1})}$
    - se string que não é codificação válida de um par grafo/inteiro, então corresponde ao par  $(G, 2)$ , onde  $G$  é o grafo trivial

# Modelo de Computação: Máquina de Turing

Modelo que usaremos nesta disciplina:

## Def.: Máquina de Turing

$M = (\Gamma, Q, \delta)$  tal que

- $\Gamma = \{\triangleright, \square, 0, 1\}$
- $Q = \{q_{\text{START}}, q_{\text{HALT}}\} \cup \{q_1, \dots, q_n\}$
- $\delta : Q \times \Gamma^3 \rightarrow Q \times \Gamma^2 \times \{L, S, R\}^3$

A respeito de MTs:

- Tipicamente com 3 fitas (entrada, trabalho, saída)
- Quando conveniente, poderemos usar mais fitas, mais símbolos, pseudo-código, etc
- Seja  $\alpha$  uma string (binária):
  - $M_\alpha$  é a MT descrita por  $\alpha$  (i.e.,  $\ulcorner M_\alpha \urcorner = \alpha$ )
- Vamos assumir que
  - Toda string represente alguma MT
  - Uma MT pode ser representada por infinitas strings