Algoritmos - Aula 12

Fernando Raposo

Vamos ver

- Complexidade Computacional
  - Conceitos Introdutórios
  - Problemas Computacionais
    - Problemas de Busca
    - Problemas de Decisão
  - Caixeiro Viajante
  - Computabilidade
  - Solubilidade
  - Problemas Classe P
  - Problemas Classe NP
  - Redutibilidade
  - P ≠ NP

Introdução

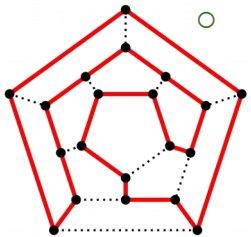
- *Você já se perguntou se existe um problema que na teoria você saberia resolver, mas na prática não seria possível?*

- Correção Algorítmica
  - Um algoritmo é dito **CORRETO** se, para cada instância, ele gera a saída correta.
  - Dizemos então que um Algoritmo Correto **RESOLVE** um dado Problema Computacional.
- Complexidade: Definição Teórica
  - **É o consumo de tempo do melhor algoritmo possível para um problema (no pior caso).**

- Mas… Já conhecemos o melhor algoritmo?
  - Podemos dizer que quase sempre o melhor algoritmo ainda é desconhecido.

Problemas Computacionais

- Lembram-se destas questões?
  - *Dados dois vértices a e b de um grafo, e um número n. Encontrar um caminho de a a b neste grafo que tenha comprimento menor que n. (Ou confirmar que não existe tal caminho)*
  - *Encontrar um ciclo hamiltoniano (ou seja, um ciclo que passe por todos os vértices) num grafo dado (ou constatar que tal ciclo não existe)*



  - *Dado um número natural n, encontrar um número natural p, maior que 1 mas menor que n, que seja divisor de n (ou constatar que tal p não existe).*
- Ou destas
  - *Menor caminho: Dados vértices a e b de um grafo, encontrar um caminho de comprimento mínimo de a a b no grafo (ou constatar que não há caminho algum de a a b)*
  - *Encontrar o MDC entre dois números.*

Problemas Computacionais

- Os problemas em verde categorizam-se como: Problemas de Busca e/ou Decisão
  - Respostas: SIM ou NÃO
- Os problemas em azul categorizam-se como: Problemas de Otimização
  - Os problemas de otimização podem ser formulados como problema de busca se consideramos que cada resposta *R* tem um custo *C* e buscamos pelas respostas que excedem um limiar ou atingem um valor máximo.

Caixeiro Viajante

- O algoritmo do Caixeiro Viajante é um problema de otimização.
- Vimos que calcular o trajeto ótimo tem um custo computacional **altíssimo** mesmo para instâncias pequenas.
  - Hipóteses: **(n-1)!**
  - Complexidade: **O(n!)**

- Logo vamos tentar caracterizar os problemas...

Função de complexidade	Tamanho da Instância do Problema					
	10	20	30	40	50	60
n	0,00001 segundos	0,00002 segundos	0,00003 segundos	0,00004 segundos	0,00005 segundos	0,00006 segundos
n^2	0,0001 segundos	0,0004 segundos	0,0009 segundos	0,0016 segundos	0,0025 segundos	0,0036 segundos
n^3	0,001 segundos	0,008 segundos	0,027 segundos	0,064 segundos	0,125 segundos	0,216 segundos
n^5	0,1 segundos	3,2 segundos	24,3 segundos	1,7 minutos	5,2 minutos	13,0 minutos
2^n	0,001 segundos	1,0 segundos	17,9 segundos	12,7 dias	35,7 anos	366 séculos
3^n	0,059 segundos	58 minutos	6,5 anos	3855 séculos	2×10^8 séculos	$1,3 \times 10^{13}$ séculos

Computabilidade

- Hoje em dia aprendemos na escola a resolver equações quadráticas: $ax^2 + bx + c = 0$ (lembram de Báskara?) $x = \frac{1}{2a}(-b \pm \sqrt{b^2 - 4ac})$
- Foram achadas fórmulas para resolver equações de grau 3 e 4, e iniciou-se a busca incansável pela ***fórmula quíntica***;
- Até que a álgebra moderna levou a uma resposta surpreendente: **Não Há!**
- Mas… qual o obstáculo para a fórmula quíntica?

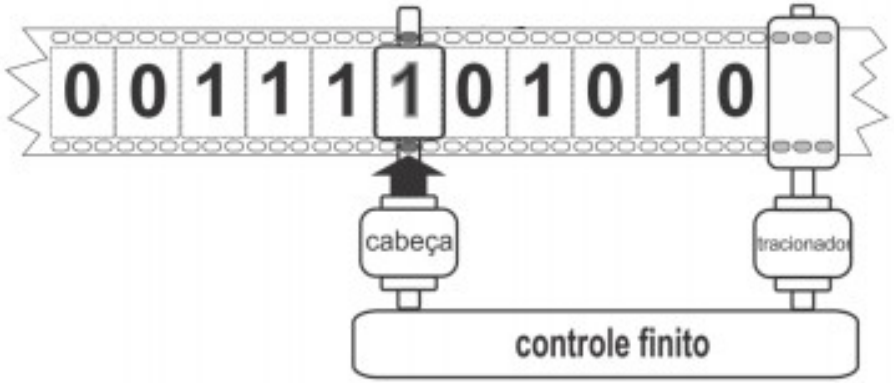
- Por que não existe fórmula de x tal que: $ax^5 + bx^4 + cx^3 + dx^2 + ex + f = 0$

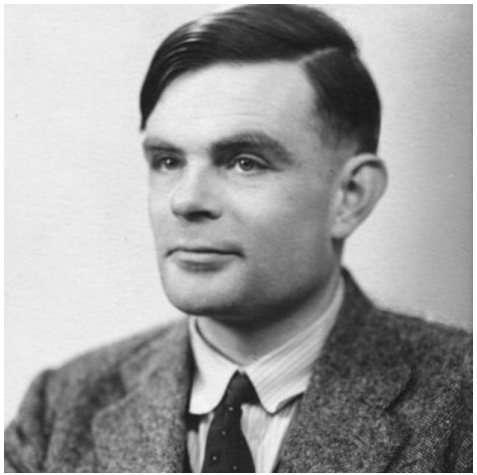
Computabilidade

- Quando falamos de fórmula, estamos falando de uma sequência de passos finita e bem determinada que deve ser seguida com cuidado para obtermos o resultado;
- Ou seja, não há algoritmo estruturado capaz de resolver a quíntica;

Computabilidade

- Um problema computável é aquele em que há uma máquina de Turing capaz de solucioná-lo.
  - **Máquina de Turing:** Uma seqüência finita de instruções, as quais podem ser realizadas mecanicamente, em tempo finito.





Solubilidade

- Problema Solucionável
  - Um problema é dito Solucionável ou Totalmente Solucionável se **existe um algoritmo** que solucione o problema tal que sempre termine para qualquer entrada, com uma resposta afirmativa (aceita) ou negativa (rejeita).
- Problema Não-solucionável
  - Um problema é dito Não-Solucionável se **não existe um algoritmo** que solucione o problema tal que sempre termine para qualquer entrada.
- Problema Parcialmente Solucionável
  - Um problema é dito Parcialmente Solucionável se existe um algoritmo que solucione o problema tal que termine quando a resposta é afirmativa (aceita).
  - Mas, quando a resposta esperada for negativa, o algoritmo pode terminar (rejeita) ou entrar em loop infinito.

Solubilidade (cont…)

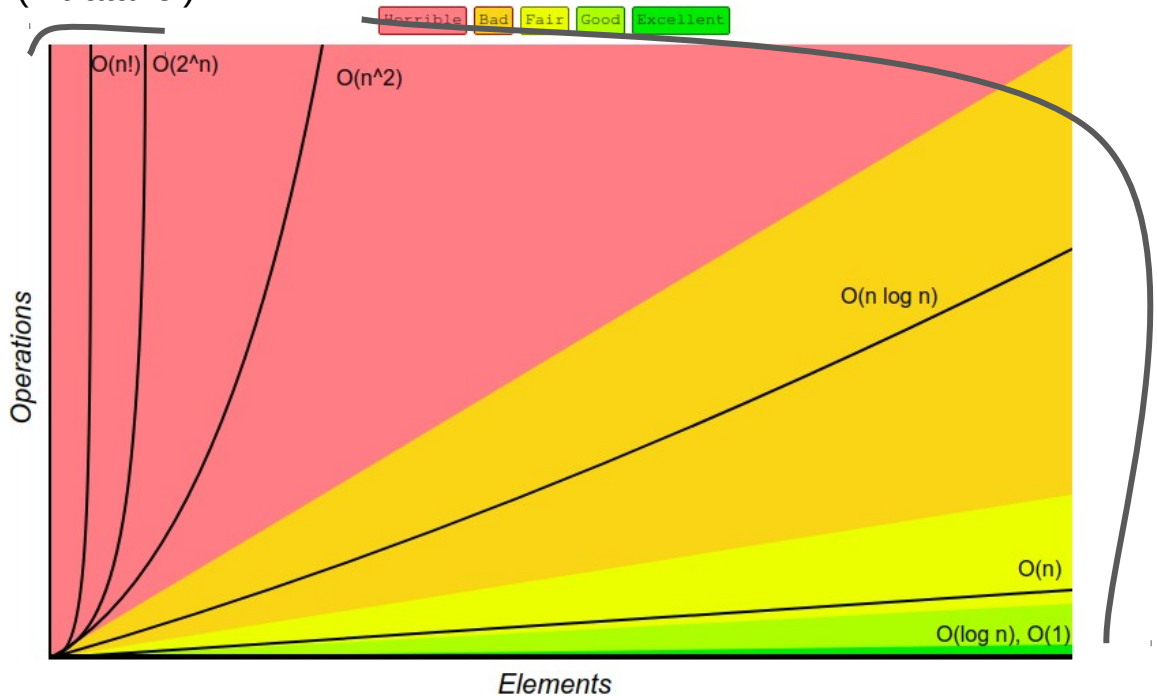
- Problema Completamente Insolúvel ou Não-Computável
  - Um problema é dito Completamente Insolúvel ou Não-Computável se não existe um algoritmo que solucione o problema tal que termine quando a resposta é afirmativa (aceita).

Complexidade

- Concluímos que o desenvolvimento de diversos algoritmos (procedimentos computacionais de passo-a-passo) nos trouxe a noção clara de **Computabilidade**;
- Entretanto, a **incomputabilidade** tornou-se uma das leis que limitam a nossa capacidade de resolver problemas;
- Há classes de problemas em que um esforço computacional consegue chegar a uma solução em tempo polinomial;
- Há classes de problemas que possuem solução, mas uma solução **custosa** (exponencial, intratável)

Possibilidade de Computabilidade

Tempo Não-Polinomial
(Intratável)



Tempo Polinomial
(Tratável)

Problemas da Classe P

- É a classe de todos os problemas de decisão que podem ser resolvidos por algoritmos polinomiais;
- São chamados de problemas tratáveis, ou problemas fáceis, cujo algoritmo é de ordem $O(n^k)$;
  - Ordenações (Mergesort, Bubblesort, Heapsort)
  - Busca Binária
  - Busca em Grafos (Profundidade, Largura, Identificar ciclos)
  - Resolver equações do segundo grau

Problemas da Classe P

- Um problema da classe P é “fácil” de resolver em tempo polinomial e “fácil” de verificar.
- Se eu recebo um array e digo que ele está ordenado, verificar a suposta ordenação é feito em tempo polinomial, assim como a implementação da ordenação em si.

Problemas da Classe NP

- **Formada pelos problemas de decisão que não podem ser resolvidos em tempo polinomial MAS possuem um verificador polinomial para a resposta SIM;**
- **NP** = "*Nondeterministic Polynomial time*" não confundir com ~~Não-Polinomial~~;

- Podemos lembrar de problemas de Criptografia, o do Caixeiro Viajante, e o Sudoku que veremos adiante.
- Vocês sabem jogar Sudoku?

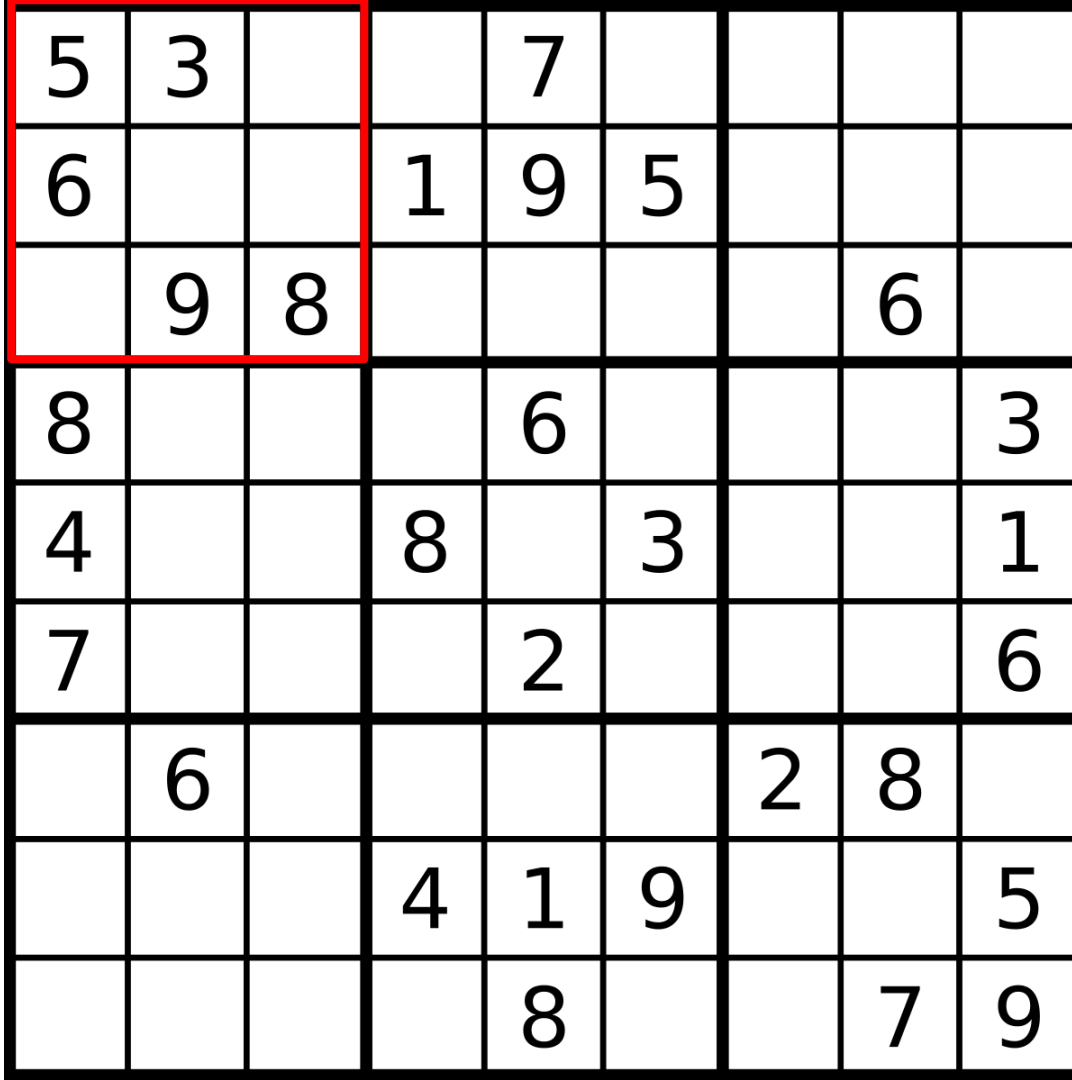
Exemplo 1

- Números de 1-9;
- Números sem repetição nos quadrados;
- Números sem repetição em cada linha ou coluna.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

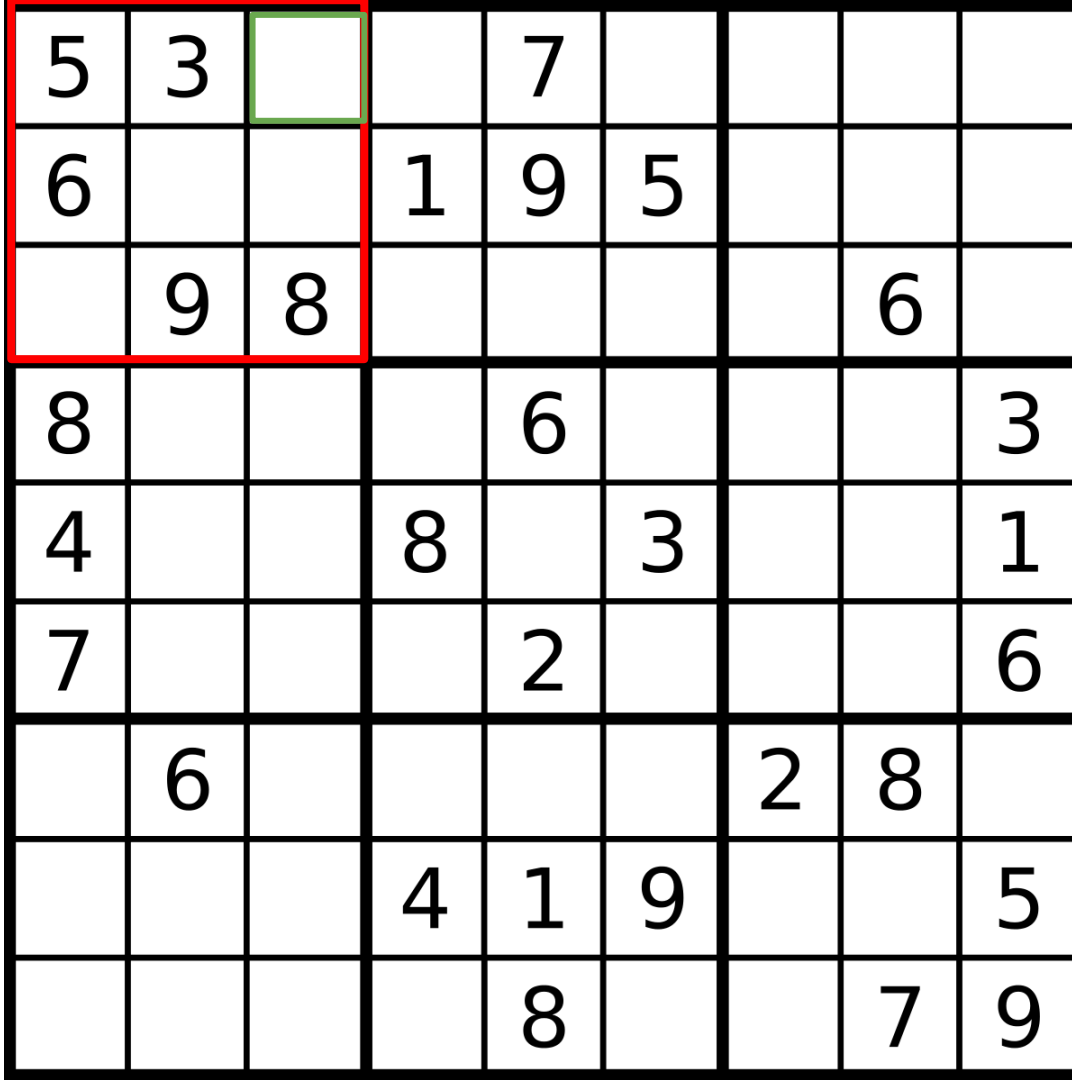
Exemplo 1

- No primeiro quadrado
  1, 2, 4 e 7 estão faltando;



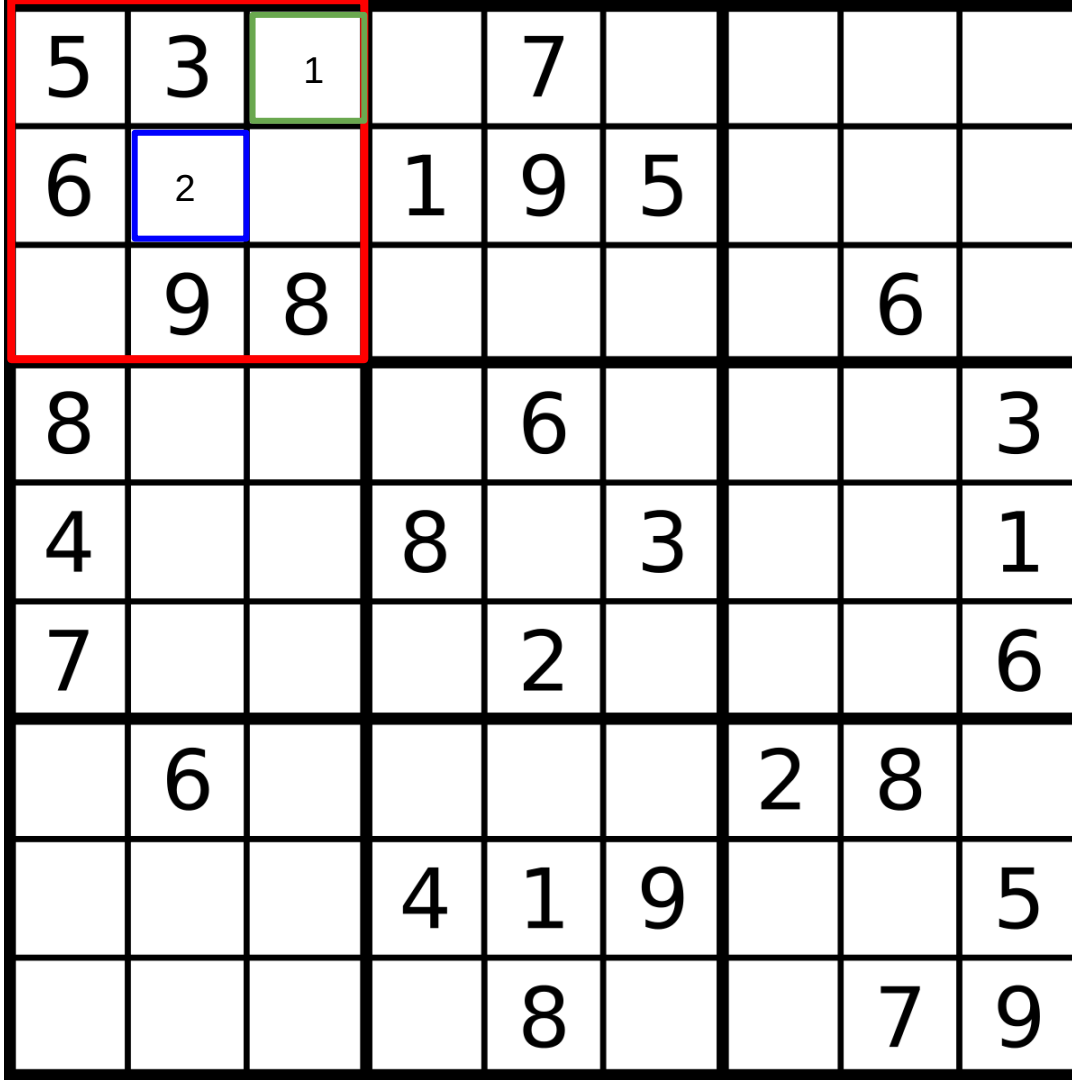
Exemplo 1

- No primeiro quadrado
  1, 2, 4 e 7 estão faltando;
- Assim, se formos começar
  com este quadrado,
  há 4 opções



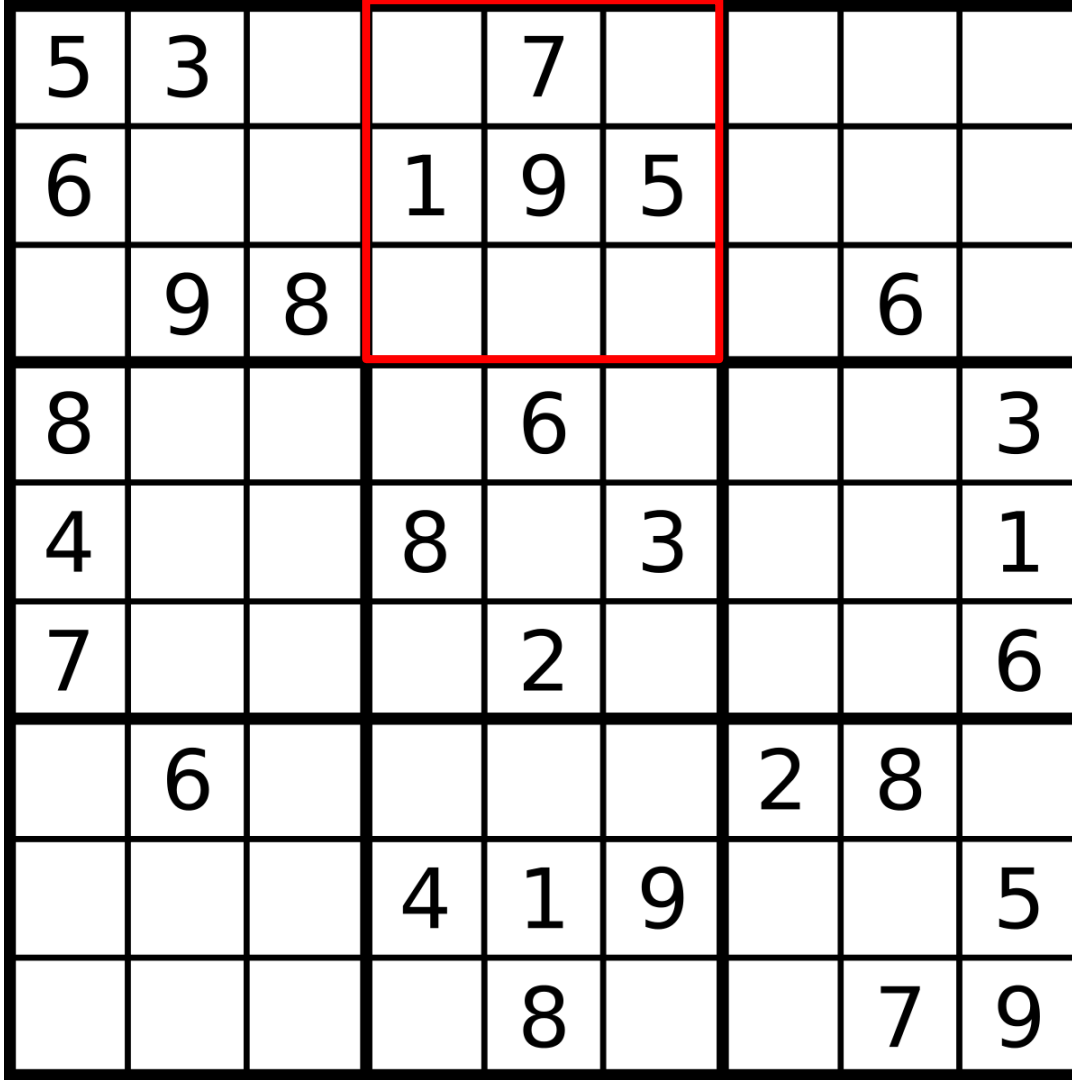
Exemplo 1

- No primeiro quadrado 1, 2, 4 e 7 estão faltando;
- Depois, temos 3 opções…
- Depois, temos 2 opções…
- No final 4! apenas para este quadrante;



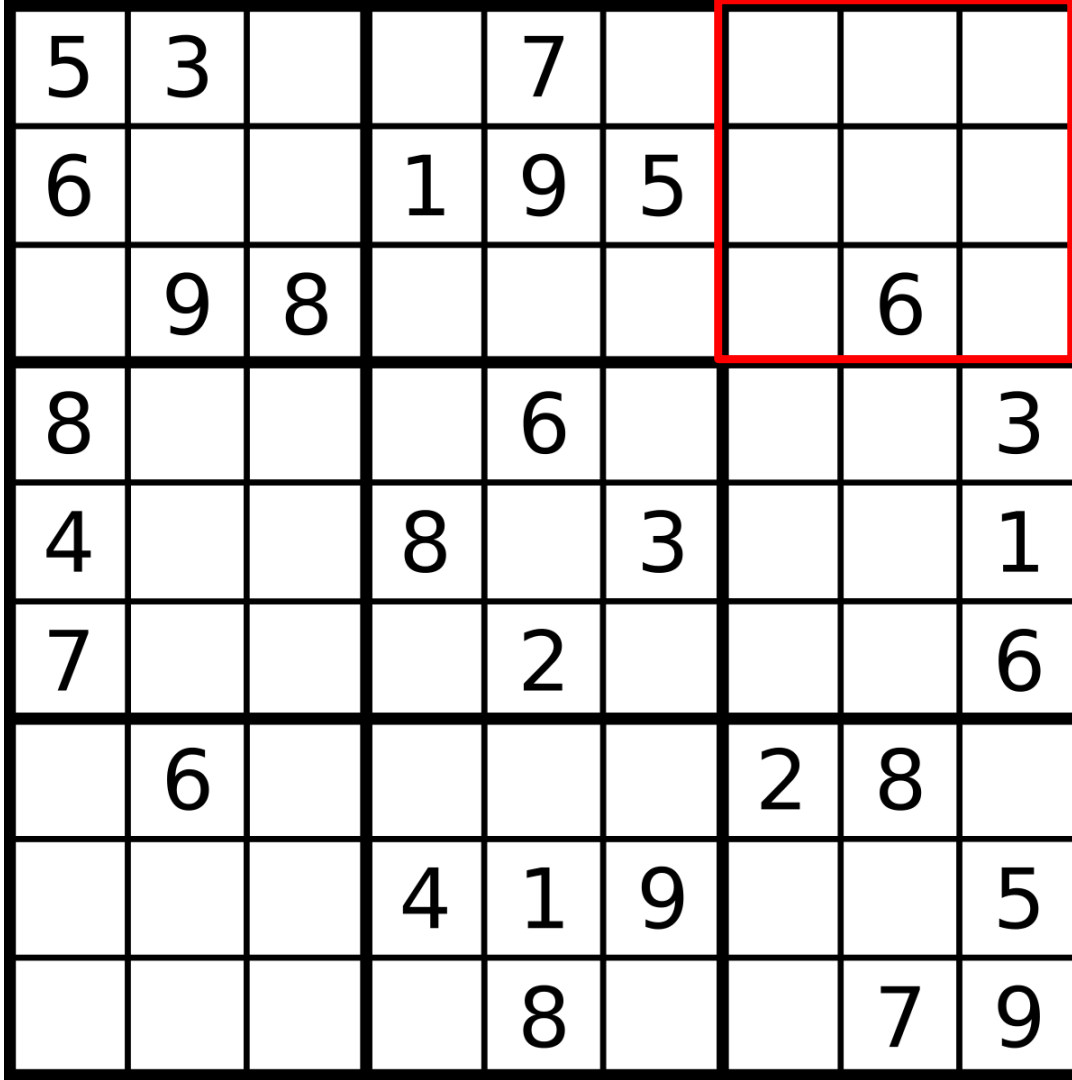
Exemplo 1

- E faremos similar no seguinte com 5!...



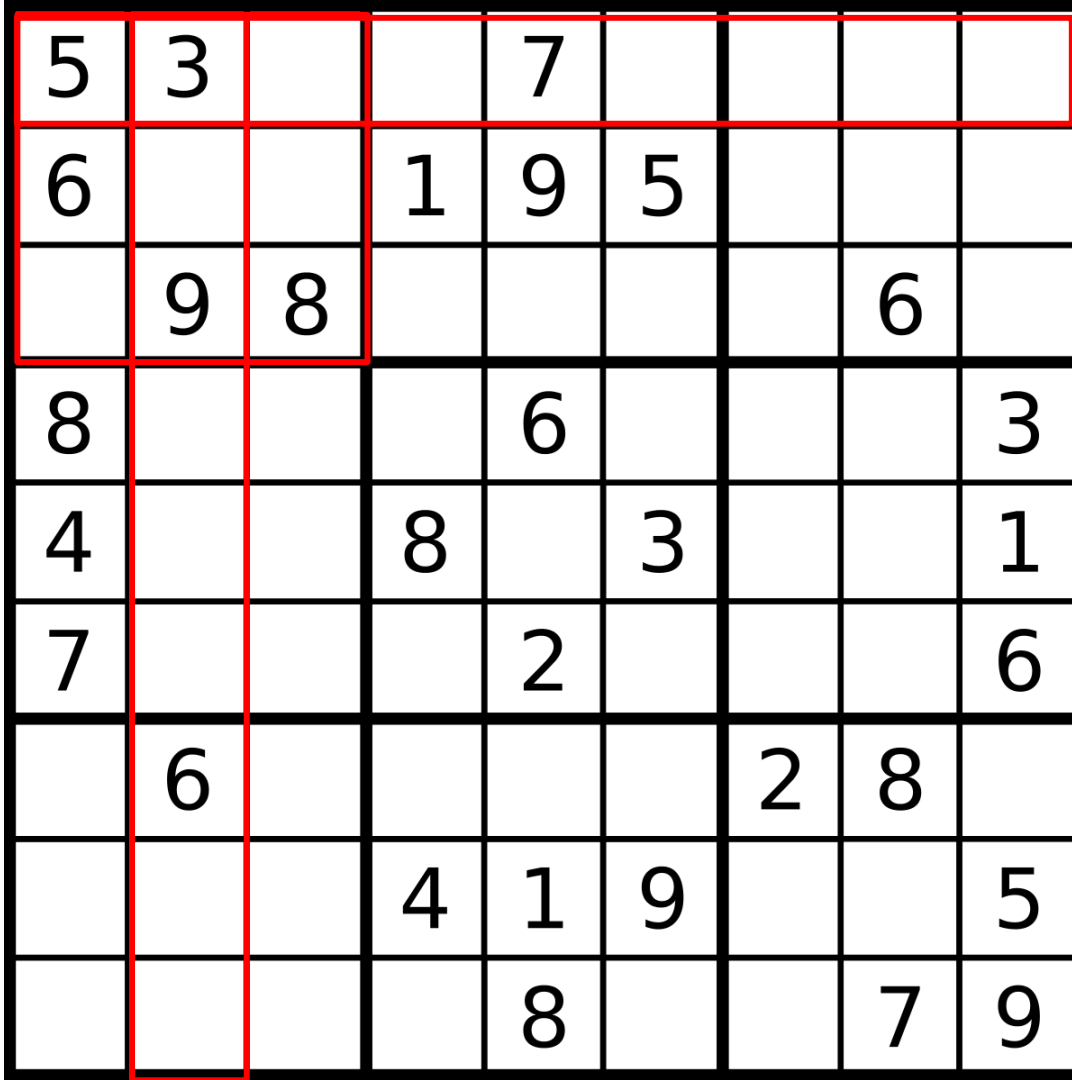
Exemplo 1

- E faremos similar no seguinte com 8!...
- Resolver isso, além de ser difícil, gasta muito tempo (Fatorial);



Exemplo 1

- Contudo, se uma pessoa vem e me diz: “Olha eu resolvi esse Sudoku!”. Eu consigo verificar linhas, colunas e quadrados em tempo polinimial



Problemas da Classe NP

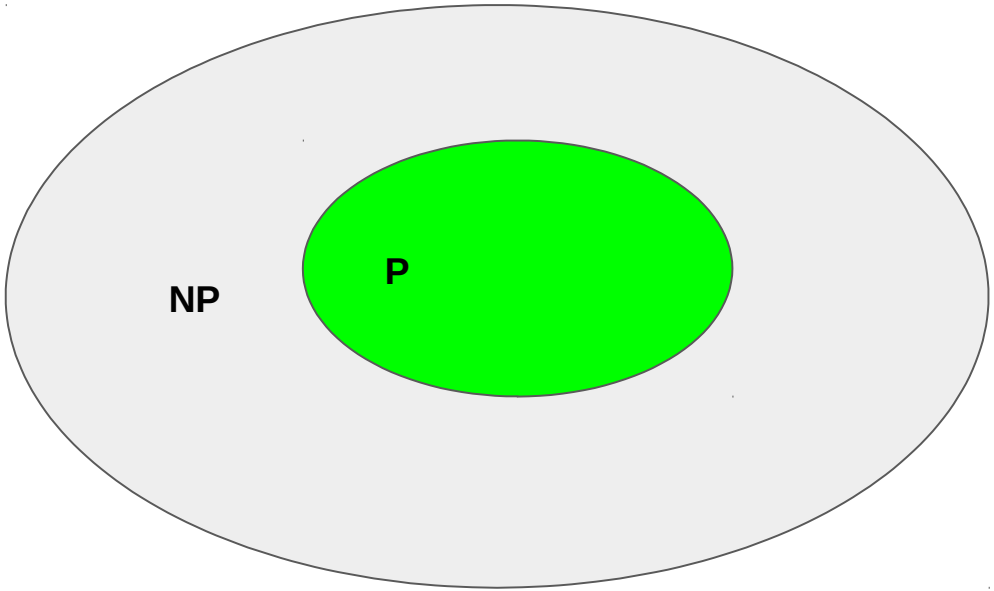
- Resumindo:
- Um problema NP é “difícil” de resolver e “fácil” de verificar.
- Exemplo 2:
  - **Equação do segundo grau:** É fácil verificar se um dado inteiro x satisfaz a equação $ax^2 + bx + c = 0$: basta calcular o número $a \times x \times x + b \times x + c$ e compará-lo com zero. Se x satisfaz a equação então o número de dígitos de x não passa do número de dígitos de (a, b, c) e portanto os cálculos consomem tempo limitado por um polinômio no números de dígitos de (a, b, c). Portanto, o problema da equação do segundo grau pertence à classe NP.

Problemas da Classe NP

- Um momento… a equação de segundo grau faz parte tanto da classe P quanto da NP?
  - Sim, pois se eu consigo resolver em tempo polinomial, eu consigo tanto verificar, quanto, dados números inteiros a, b e c, encontrar um número inteiro x tal que $ax^2 + bx + c = 0$ (ou constatar que tal x não existe).

- Portanto: $\mathbf{P \subseteq NP}$
- Ou seja, se um algoritmo pode ser resolvido em tempo polinomial, ele também pode ser verificado em tempo polinomial.

Problemas da Classe NP



P ≠ NP ou P = NP?

- Qual a implicação disto? O que aconteceria se uma pessoa conseguisse provar que P = NP?
- Questão de 1 milhão de dólares;

- Já se você provar que P ≠ NP, você vai demonstrar que existe uma classe de problemas que não tem solução.

P ≠ NP ou P = NP?

- Toda a segurança da informação (algoritmos de criptografia) e de segurança na web tornariam-se vulneráveis a ataques;
- TUDO se tornaria mais eficiente, tal como transportes, entendimento de DNA, Bitcoin...
- Contudo até hoje ninguém encontrou ainda um problema de NP que não esteja em P, isto é, um problema polinomialmente verificável para o qual não existe algoritmo polinomial.

Redutibilidade

- Sabemos que o status de muitos problemas em NP é incerto, portanto é bom verificar a sua complexidade relativa;
- Informalmente, dizemos que um problema Y *não é mais difícil* que um problema X se Y for um subproblema, ou caso particular, de X

Redutibilidade

- É comum na resolução de um problema “Y”, reduzi-lo a um problema “X”;
- Quadrado Perfeito:
  - Dado um número natural n, encontrar um número natural x tal que $x^2 = n$ (ou constatar que tal x não existe).
- Equação do segundo grau:
  - Dados números inteiros a, b e c, encontrar um número inteiro x tal que $ax^2 + bx + c = 0$ (ou constatar que tal x não existe)

Assim, vemos que o problema do quadrado perfeito é um caso particular, ou um subproblema da equação do segundo grau.

- Ou seja, o problema do quadrado perfeito não é mais difícil (ou pode ser reduzido) que o problema da equação do segundo grau.

Redutibilidade

- Portanto entendemos que:

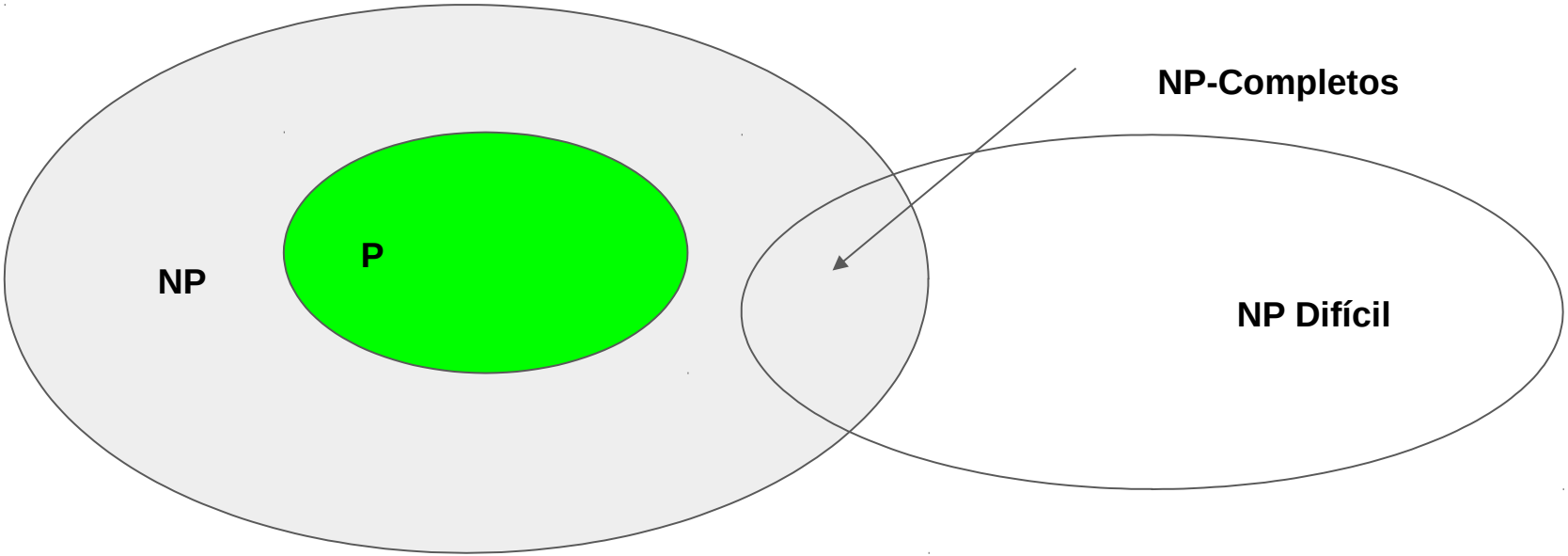
“Se Y não é mais difícil que X e X está na classe P então Y também está na classe P”

ou

““Se Y é redutível a X e X está na classe P então Y também está na classe P”

Problemas Classe NP-Completo e NP-Difícil

- Um problema X é NP-Difícil se todos os problemas em NP são redutíveis a ele em tempo polinomial;
- Um problema é NP-Completo se for NP-Difícil e estiver em NP;




Thank You!