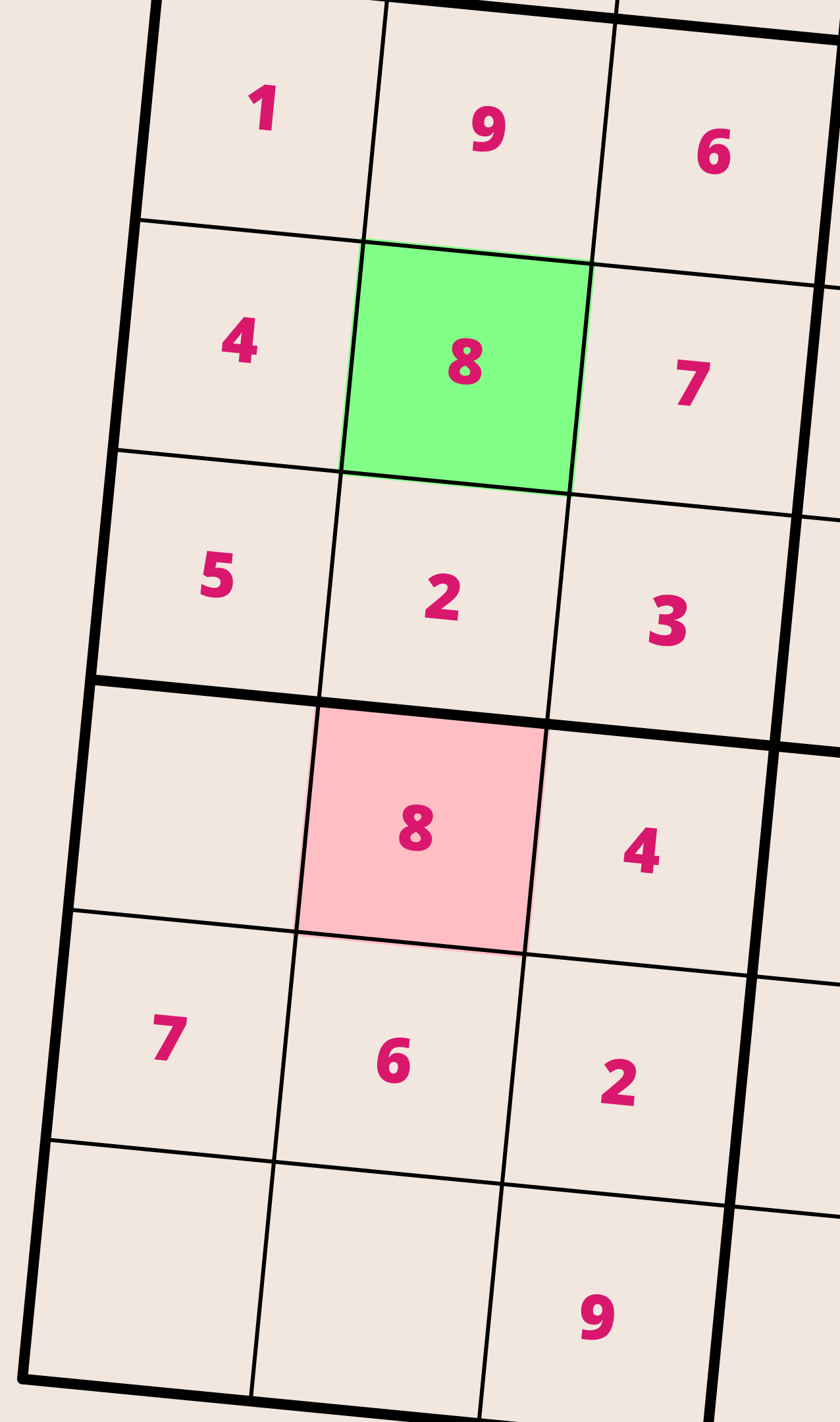


Let's Play Sudoku!



Autores: Akles Pires Camoleze e João Felipe Silva Lamounier



Sudoku com Backtracking em Python

Dê um passo adiante na resolução de quebra-cabeças com o nosso solucionador de Sudoku baseado em Python! Nesta apresentação minimalista, mergulhe nas profundezas do algoritmo de backtracking que torna possível desvendar até os Sudoku mais desafiadores.



S

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

O

U

K

D

U

NÚMEROS POSSÍVEIS PARA PRIMEIRA POSIÇÃO{3,5,6,7,8,9}

S

O

U

K

D

U

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

NÚMEROS POSSÍVEIS PARA PRIMEIRA POSIÇÃO{3,6,7,9}

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

S

O

U

K

D

U

INSERINDO A PRIMERIA LINHA

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

S

O

U

K

D

U

S

O

U

K

D

U

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

S

U

D

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



NÃO É POSSÍVEL ADICIONAR MAIS NÚMEROS NESTA POSIÇÃO, PORTANTO, EMPREGAMOS A TÉCNICA DE "BACKTRACKING" PARA RETROCEDER À POSIÇÃO ANTERIOR E TENTAR INSERIR O PRÓXIMO NÚMERO VÁLIDO, COM BASE NOS ELEMENTOS DISPONÍVEIS.

K

U

S

U

D

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



NESTA POSIÇÃO OCORRE O MESMO PROBLEMA, PORTANTO CONTINUAMOS COM O, “BACKTRACKING”, ATÉ SURGIR NOVAS OPÇÕES.

K

U

INSERÇÃO FINAL

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

S

O

U

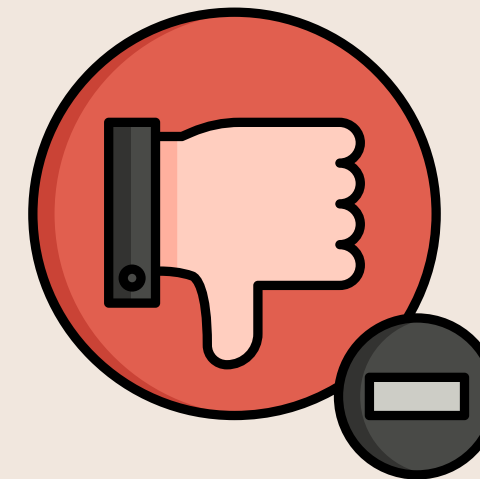
K

D

U



Vantagens X Desvantagens do Backtracking



Garantia de solução: O algoritmo de backtracking garante que uma solução será encontrada, desde que haja uma solução válida para o Sudoku em questão. Isso elimina a incerteza em relação à resolução do quebra-cabeça.

Facilidade de implementação: O algoritmo de backtracking é relativamente simples de implementar, tornando-o acessível mesmo para iniciantes em programação.

Algoritmo Determinístico: O backtracking é um algoritmo determinístico, o que significa que ele sempre seguirá o mesmo conjunto de passos para resolver o Sudoku. Isso o torna previsível e útil para fins educacionais, pois é fácil de entender e explicar como funciona.

Complexidade temporal: A complexidade temporal do backtracking pode ser alta em Sudoku, quando muitas opções estão disponíveis em cada célula, levando a um tempo de execução mais longo.

Uso de memória: Em Sudoku muito grandes, o backtracking pode consumir uma quantidade significativa de memória à medida que mantém o histórico das decisões tomadas em cada nível de recursão.

Incapacidade de encontrar múltiplas soluções: O backtracking geralmente encontra uma única solução para um Sudoku, mesmo que haja várias soluções possíveis. Isso pode ser uma desvantagem se você estiver interessado em encontrar todas as soluções ou se estiver usando Sudoku como parte de um quebra-cabeça mais complexo.

