

Contextualização:

Você trabalha na *Ximquiling Entertainment Games*, uma importante empresa multinacional da área de jogos eletrônicos. O presidente da *Ximquiling*, com a intenção de motivar e avaliar seus funcionários lançou um desafio na empresa, um concurso de algoritmos para resolver probleminhas de Sudoku. Segundo ele, o funcionário que conseguir desenvolver o algoritmo mais rápido, ou seja, aquele que obtém a solução de um Sudoku em menor tempo, terá um aumento salarial de 300% e se tornará o vice-presidente da empresa.

A palavra Sudoku significa “número sozinho” em japonês, o que mostra exatamente o objetivo do jogo. O Sudoku existe desde a década de 1970, mas começou a ganhar popularidade no final de 2004 quando começou a ser publicado diariamente na sessão de *puzzles* do jornal *The Times*. Entre abril e maio de 2005 o *puzzle* começou a ganhar um espaço na publicação de outros jornais britânicos e, poucos meses depois, ganhou popularidade mundial.

O objetivo do Sudoku é preencher todos os espaços em branco do quadrado maior, que está dividido em nove grids, com os números de 1 a 9. Os algarismos não podem se repetir na mesma coluna, linha ou grid.

A figura a seguir mostra um exemplo de Sudoku e sua solução.

Instância

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

Solução

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

Objetivo:

Este trabalho tem por objetivo, exercitar os conceitos relacionados aos paradigmas de projeto de algoritmos, mais especificamente o Backtracking para solução de problemas de sudoku.

Para esse trabalho, elabore uma estratégia, usando backtracking, para solucionar problemas de Sudoku.

Para o runcodes você deve imprimir somente a saída. Para apresentação imprima tanto a entrada quanto a saída (solução) do sudoku.

Para o exemplo da figura acima, o arquivo de entrada seria como apresentado abaixo: onde o 0 (zero) representa uma casa vazia.

Exemplo de entrada	Exemplo de saída
5 3 0 0 7 0 0 0 0	-----
6 0 0 1 9 5 3 4 8	5 3 4 6 7 8 9 1 2
0 9 8 0 0 0 0 6 7	6 7 2 1 9 5 3 4 8
8 0 0 0 6 0 0 0 3	1 9 8 3 4 2 5 6 7
4 0 0 8 0 3 0 0 1	-----
7 0 0 0 2 0 0 0 6	8 5 9 7 6 1 4 2 3
0 6 0 0 0 0 2 8 4	4 2 6 8 5 3 7 9 1
0 0 0 4 1 9 0 0 5	7 1 3 9 2 4 8 5 6
0 0 0 0 8 0 0 7 9	-----
	9 6 1 5 3 7 2 8 4
	2 8 7 4 1 9 6 3 5
	3 4 5 2 8 6 1 7 9

Entrega:

O que deve ser entregue?

1. Um arquivo com o **código** do sistema em linguagem C (pelo run codes).
2. Um arquivo em pdf com o **relatório** que deve conter *i)* a contextualização do problema, *ii)* o método de solução, como é a proposta de solução desenvolvida, *iii)* estruturas de dados que foram utilizadas, *iv)* fluxograma do algoritmo, *v)* pseudo-código com descrição das principais rotinas, *vi)* resultados, e *vii)* conclusão.
3. Para os testes realizados (resultados) mostre a contagem das operações e o tempo total em milisegundos gasto para executar. Apresente 2 gráficos um “número de operações × instância” e o outro “tempo de execução × instância”.
4. Uma apresentação do trabalho de aproximadamente 10 a 15 min será realizada em laboratório.

Dica! Comece a fazer o trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar!
