



Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Departamento de Ciências de Computação

SCC0222 – Laboratório de Introdução à Ciência da Computação I

## Trabalho 05: Sudoku $16 \times 16$

**Professor:** Dr. Rodrigo Fernandes de Mello (mello@icmc.usp.br)  
**Estagiário PAE:** Fábio Henrique Gomes Sikansi (fhenrique@usp.br)  
Martha Dais Ferreira (daismf@icmc.usp.br)  
**Monitor:** Lucas Parras (parraslucas@gmail.com)  
Loys Gibertoni (loys.gibertoni@usp.br)  
**Colaborador:** Felipe Simões Lage Gomes Duarte (fgduarte@icmc.usp.br)

### 1 Objetivo do Trabalho

Você deverá implementar um sistema capaz de resolver um sudoku  $16 \times 16$ .

### 2 Sudoku

Sudoku é um quebra-cabeça baseado na colocação lógica de números. O objetivo do jogo é a colocação de números de 1 a 9 em cada uma das células vazias numa grade de  $9 \times 9$ , constituída por  $3 \times 3$  subgrades chamadas regiões. O quebra-cabeça contém algumas pistas iniciais, que são números inseridos em algumas células, de maneira a permitir uma indução ou dedução dos números em células que estejam vazias. Cada coluna, linha e região só pode ter um número de cada um dos 1 a 9. Resolver o problema requer apenas raciocínio lógico e algum tempo.

A atração do jogo é que as regras são simples, contudo, a linha de raciocínio requerida para alcançar a solução pode ser complexa. O sudoku é recomendado por alguns educadores como um exercício para o pensamento lógico. O nível de dificuldade pode ser selecionado para combinar com o público. Existem diversas fontes na internet não ligadas a editoras que disponibilizam os jogos gratuitamente.

Seu formato é mais frequentemente uma grade de  $9 \times 9$  constituída de sub-grades de  $3 \times 3$  chamadas de “regiões” (outros termos incluem “caixas” e “blocos”; algumas vezes, o termo “quadrante” é utilizado, apesar de ser um termo impreciso para uma grade de  $3 \times 3$ ). Algumas células já contém números, chamadas “pistas”. **O objetivo é preencher as células vazias com um número em cada célula, de maneira que cada coluna, linha e região contenha os números de 1 a 9 apenas uma vez. Na solução do jogo, cada número aparece apenas uma vez em qualquer um dos sentidos ou regiões; daí, o termo sudoku, que significa “únicos números”.**

Apesar disso, o número de soluções de sudoku para uma grade-padrão de  $9 \times 9$  foi calculado em 2005 por Bertram Felgenhauer como sendo 6670903752021072936960. Este número é igual a  $9! \times 72^2 \times 2^7 \times 27.704.267.971$ , o último fator o qual é um número primo. O resultado é derivado através da lógica e computação força bruta. A derivação deste resultado foi simplificada consideravelmente por análises fornecidas por Frazer Jarvis e o número foi confirmado independentemente por Ed Russell. Russel e Jarvis também demonstraram de que quando as simetrias são levadas em conta, havia 5 472 730 538 soluções. O número de soluções válidas para a variação do sudoku de uma grade  $16 \times 16$  é desconhecido.

Em nosso contexto, utilizaremos um tabuleiro  $16 \times 16$  preenchido com números hexadecimais tal como a Figura 1(a). O seu programa deve preencher as células vazias tal como a Figura 1(b)

		A			F				C	B			5		
						6	7	4	3		2	B			
				E			2	4	8				0		F
5	8	9	0		D										1
	9						3	6			7	E			8
					D			9		0	6				3
0	6		B			E		D						C	
2	5	F			4			C		A					
									4				F	B	
		D				4			5	A			8	2	
C	A	5		8	2		3	9		D	F		7		6
		8		E	0	D				3	B	A			4
						8		B		C					
	E				5			1					4		2
1		3	6	B											
7	2		A		E	C						D			

(a) Posicionamento Inicial

4	3	A	2	9	F	1	E	0	C	B	D	8	5	6	7
F	C	1	D	0	8	6	7	4	3	5	2	B	A	E	9
B	7	6	E	5	3	2	4	8	9	1	A	0	D	F	C
5	8	9	0	C	D	A	B	F	6	E	7	3	2	4	1
D	9	4	C	F	A	3	6	5	B	7	E	2	1	8	0
A	1	E	8	D	B	5	9	2	0	6	C	4	F	7	3
0	6	7	B	2	1	E	8	D	F	4	3	5	C	9	A
2	5	F	3	7	4	0	C	1	A	9	8	6	E	D	B
E	0	2	9	A	C	7	5	6	4	8	1	F	B	3	D
3	B	D	1	6	9	4	F	7	5	A	0	C	8	2	E
C	A	5	4	8	2	B	3	9	E	D	F	1	7	0	6
6	F	8	7	E	0	D	1	C	2	3	B	A	9	5	4
9	D	0	5	1	6	8	2	B	7	C	4	E	3	A	F
8	E	C	F	3	5	9	D	A	1	0	6	7	4	B	2
1	4	3	6	B	7	F	A	E	D	2	5	9	0	C	8
7	2	B	A	4	E	C	0	3	8	F	9	D	6	1	5

(b) Resultado Final

Figura 1: Tabuleiro do Sudoku  $16 \times 16$ 

### 3 Proposta

O sistema deve ser capaz de resolver um sudoku  $16 \times 16$  e escrever no stdout o resultado final. Para isso, seu programa receberá como entrada somente o nome do arquivo:

teste.mat

Seu programa deve abrir este arquivo como leitura e carregar o tabuleiro para a memória do computador. O arquivo contém uma matriz  $16 \times 16$  de char tal como o exemplo a seguir:

```

..687.D.3E.9..C.
7E0.5.2.8....B..
5D.B.3A.427.E.8.
.....FC0.....4
CB46...02.F.8.7.
...FA..5D...9...
D....E.15.C....6
.....4CE.D
.08.2F1.....3.9B
..F5.6...3B7....
..7...E9...814..
..E...0...2..85.
.....F..D46A.
6.....9..5.1...3
AF3..DC.EB..7...
8....4.3...A.9.0

```

O seu programa deverá imprimir o resultado final tal como o exemplo abaixo:

```

F46870DB3E5921CA
7E0C51248ADF6B39
5D1B93AC4276E08F
3A92E86FC01B57D4
CB46D93021FE8A75
E72FACB5D6809341
D8A94EF157C302B6
03516782B9A4CEFD
408A2F176DE53C9B
29F5C64813B7AD0E
BC7D35E9AF081462
16E3BA0D942CF857
95B0127EF83D46AC
62C78B9A0541DFE3
AF340DC6EB927518
81DEF4537C6AB920

```

## 4 IMPORTANTE

- É obrigatório o uso de alocação dinâmica para qualquer vetor/matriz utilizado em seu programa.
- É obrigatório a criação das seguintes funções:
  1. `char **allocMat(int c, int r)` - Função que recebe o numero de colunas `c` e linhas `r` e retorna uma matriz alocada dinamicamente.
  2. `char **readMat(char *path, int *c, int *r)` - Função que recebe uma String com o nome do arquivo, e retorna a matriz de caracteres alocada e preenchida, o número de colunas `c` e linhas `r` lida.
  3. `void printMat(char **mat, int c, int r)` - Função que recebe uma matriz `mat`, o número de colunas `c` e linhas `r` e imprimir de acordo com o exemplo de output apresentado anteriormente.

Estas funções são obrigatórias e não podem ser alteradas, i.e., o número de parâmetros e a ordem deles deve ser tal como está descrito acima. Nada impede a criação de outras funções caso exista necessidade.