

Puzzle - Sudoku

Este exercício consiste no desenvolvimento de um algoritmo capaz de resolver um jogo de Sudoku, caso as combinações dos valores de entrada sejam válidas e possibilitem sua execução.

Sudoku, por vezes escrito Su Doku (数独, sūdoku) é um jogo 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 9x9, constituída por 3x3 subgrades chamadas regiões (conforme ilustrado na Figura 1). 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. Os problemas são normalmente classificados em relação à sua realização. O aspecto do sudoku lembra outros quebra-cabeças de jornal. Foi criado por Howard Garns, um projetista e arquiteto de 74 anos aposentado.

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

Figura 1 – Puzzle Inicial com as Pistas VS Resultado final do Puzzle

Funcionamento

Os numerais no jogo sudoku são usados por comodidade; as relações aritméticas entre numerais são absolutamente irrelevantes (não requer lógica para cálculos matemáticos). Qualquer combinação de símbolos distintos como letras, formas, ou cores podem ser usadas no jogo sem alterar as regras. Por exemplo, algumas variações usam letras, como Scramblets da Penny Press e Sudoku Words da Knight Features Syndicate. Dell Magazines, o criador do jogo, tem utilizado números para Number Place em suas revistas desde a sua primeira publicação em 1979. Numerais são utilizados através deste artigo.

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

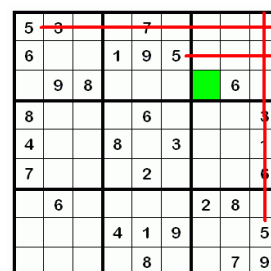
alguns educadores como um exercício para o pensamento lógico. O nível de dificuldade pode ser selecionado para combinar com o público. Existem diversas fontes na internet não ligadas a editoras que disponibilizam os jogos gratuitamente.

Seu formato é mais frequentemente uma grade de 9x9 constituída de sub-grades de 3x3 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 3x3). Algumas células já contém números, chamadas "números dados" (ou, algumas vezes, "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".

Métodos de Solução

A estratégia para a resolução de um sudoku pode ser considerada como uma combinação de três processos: fazer uma varredura visual, fazer marcações e análise.

Algumas estratégias de resolução do puzzle são detalhadas no link <https://goo.gl/1dk4dU>.



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

Objetivo do Trabalho

Desenvolver um programa, aplicando os conceitos de algoritmos vistos em sala de aula, que recebe como entrada os “números dados” de um jogo de Sudoku e execute o jogo inferindo as demais células ainda não preenchidas, baseado nas dicas (números dados) e nos requisitos do trabalho detalhados abaixo.

Requisitos do Trabalho

O jogo deverá atender aos seguintes requisitos:

1. O sistema deverá receber um único arquivo de entrada usado para o processamento do programa, denominado: **tabuleiro_dado.txt**, e deverá gerar um único arquivo de saída com o nome **resultado.txt** (ambos devem ser criados e armazenados na mesma pasta onde o programa será salvo);
2. O arquivo de entrada deverá ter o seguinte formato, simulando o tabuleiro com “números dados” da Figura 1 (ver Figura acima):

```
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

```

Sendo:

<numero> → número dado usado como dica para o desenvolvimento do *puzzle*;

→ célula do tabuleiro do puzzle sem valor preenchido (a ser computado pelo algoritmo); e

; → separador dos elementos do tabuleiro

3. As entradas serão compostas por elementos que representam uma matriz 9x9, da mesma forma que a matriz usada no puzzle, sem exceder esse número de elementos. Isso é uma premissa da entrada do trabalho!
4. Algumas validações deverão ser aplicadas sobre os dados inicialmente carregados do arquivo de entradas antes do processamento do jogo, considerando a sua respectiva ordem de precedência, a citar:

a) **Validação 1** – Não deve haver elementos repetidos na mesma região do tabuleiro do *puzzle* (matriz 3x3) – **Mensagem de erro:**

REGION_WITH_REPEATED_VALUE;

b) **Validação 2** – Não deve haver elementos repetidos na mesma linha do tabuleiro do puzzle (matriz 9x9) – **Mensagem de erro:**

LINE_WITH_REPEATED_VALUE;

c) **Validação 3** – Não deve haver elementos repetidos na mesma coluna do tabuleiro do puzzle (matriz 9x9) – **Mensagem de erro:**

COLUMN_WITH_REPEATED_VALUE; e

d) **Validação 4** – Não deve haver elementos com valores numéricos fora da faixa de valores válidos, caracteres (a-z|A-Z) e/ou símbolos especiais no tabuleiro do puzzle (matriz 9x9), apenas números que variem de 1 à 9 e o caractere “#” indicando que a célula não tem valor preenchido – **Mensagem de erro:**

INVALID_VALUE_ON_THE_BOARD.

5. Caso uma das validações acima, e na ordem de precedência mencionada, identifique alguma inconsistência no arquivo, o programa deverá escrever no arquivo **resultado.txt** a mensagem no seguinte formato:

a) <linha>|<coluna>:<mensagem_erro> - Exemplo: 2|5:

INVALID_VALUE_ON_THE_BOARD.

b) **Observação:** Apesar de matrizes e vetores computacionalmente terem como referência para o primeiro índice o valor 0 (zero), para

as saídas a serem fornecidas assumiremos que o índice referente à primeira coluna será 1 (um), variando de 1 à 9 no padrão de saída acima descrito.

6. Caso o processamento do algoritmo ocorra com sucesso, e o programa compute corretamente os valores do tabuleiro, o seguinte padrão de saída deverá ser escrito no arquivo **resultado.txt**:
- a) **Premissa:** Caso todas as validações mencionadas no item 4 sejam satisfeitas, assume-se como premissa que os “números dados” informados são suficientes para a execução do puzzle com resultado final válido.

b) **Saída:**

```
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
```

Critérios de Avaliação:

1. Programa deve ser entregue no formato de arquivo salvo pela ferramenta Portugol Studio (um único arquivo contendo o código fonte do programa) e funcional. Programas com falha de execução serão avaliados com nota 0,0;
2. Atendimento aos requisitos de **validações** descritos nos requisitos do trabalho, totalizarão 40% da nota final do trabalho;
3. Atendimento aos requisitos de processamento do jogo, inferindo os demais valores do *puzzle*, conforme descrito nos requisitos do trabalho, totalizarão 60% da nota final do trabalho;
4. Trabalhos totalmente ou parcialmente plagiados, contendo fragmentos de código copiados de outras duplas, terão suas notas divididas entre as duplas envolvidas, independente do número de duplas que efetivou a réplica; e
5. Trabalhos totalmente ou parcialmente plagiados, contendo fragmentos de código copiados de fontes externas como (artigos, web e outros), terão o decréscimo de 1,0 pontos por fragmento plagiado identificado.

Observação: A discussão das ideias de resolução do trabalho pode ser feitas entre as duplas de trabalho, bem como códigos de fontes externas podem ser analisados sem problemas. **Todavia, a implementação final deve ser autêntica e exclusiva da DUPLA definida em sala para o desenvolvimento do trabalho final.**