

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE CENTRO DE TECNOLGIA DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO



SUDOKU PROFESSOR: ADELARDO ADELINO DANTAS DE MEDEIROS

O objetivo é desenvolver em C++ um programa capaz de permitir que o usuário preencha tabuleiros de Sudoku manualmente, utilizando uma interface amigável, baseada em Qt (widgets e menus). O aplicativo também deverá ser capaz de, quando solicitado, resolver o tabuleiro automaticamente ou indicar que não existe solução. A resolução automática deverá ser feita utilizando as estruturas de dados da biblioteca STL de C++.

ALGORITMO

O algoritmo em pseudocódigo listado a seguir pode ser útil no desenvolvimento do resolvedor automático de Sudoku. Ele resolve o problema utilizando uma técnica de busca em profundidade conhecida como backtrackina.

```
// Cria conjunto inicial vazio
Conjunto<Sudoku> Aberto
// Inicializa o conjunto contendo
// apenas o tabuleiro inicial
inserir(Aberto, tab inicial)
num testados \leftarrow 0
encontrou solucao \leftarrow false
// Iteração: repita enquanto houver
// tabuleiros ainda não testados
// em Aberto e ainda não houver
// encontrado a solução
Repita
| tab ← extrai elemento(Aberto)
| num testados++
| // Testa se é solução
| Se ( é solução(tab) )
| | encontrou solucao ← true
| Caso contrário
|  | pos ← prim casa vazia(tab)
| | Para val de 1 a 9
| | | // a casa pos do tabuleiro
| | | // tab com o valor val
| | | | // Gera novo tabuleiro:
| | | | // sucessor de tab fazendo
```

```
| | | | // valor de pos igual a val
| \ | \ | \ |  suc \leftarrow faz jogada(tab,
                        pos, val)
| | | | // Insere sucessor em
 | | | inserir (Aberto, suc)
| | Fim Para
| Fim Se
| // Caso queira exibir o estado
| // parcial do algoritmo:
| exibir(tab)
| exibir(num testados)
| exibir(tamanho(Aberto))
Enquanto ( not(encontrou solucao)
           and not(vazio(Aberto)) )
// Pode ter terminado porque
// encontrou a solução ou porque
// não há mais tabuleiros a testar
Se ( not(encontrou solucao) )
| exibir("Sudoku sem solução")
| exibir(num testados)
Caso contrário
| exibir("SOLUÇÃO:")
| exibir(tab)
| exibir(num testados)
| exibir(tamanho(Aberto))
Fim Se
```

A tabela a seguir apresenta uma evolução das variáveis do algoritmo em uma situação envolvendo um tabuleiro reduzido de Sudoku (blocos de 2×2, ao invés do bloco normal de 3×3). Também é apresentado um fluxograma dos tabuleiros com seus sucessores.



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE CENTRO DE TECNOLGIA DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO



Tabuleiro	Aberto (após inclu	Nº de	
analisado	do tabuleiro se	sucessores	
-	4 3		-
1 1 3	1 2 3 2 1 1 4 3 4 3		2
1 2 1	1 2 3 1 1 1 1 1 1 4 3	3 2 1 4 3	2
1 2 3	1 2 3 4 1 2 4 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	3 2 1	1
1 2 3 4	1 2 3 4 3 1 1 1 4 3 4 3	3 2 1 4 3	1
1 2 3 4 3 1	1 2 3 4 3 4 1 1 4 3 4 3	3 2 1 4 3	1
1 2 3 4 3 4 1 4 3	1 2 3 4 3 4 1 2 1 4 3 4 3	3 2 1 4 3	1
1 2 3 4 3 4 1 2 4 3	1 2 3 4 3 4 1 2 2 1 4 3 4 3 4 3	3 2 1	1
1 2 3 4 3 4 1 2 2 4 3	1 2 3 4 3 4 1 2 2 1 2 3 4 4 3 4 1 2 2 3 4 4 3 4 1 2	1 2 4 3 2 1	2
1 2 3 4 3 4 1 2 2 1 4 3	1 2 3 4 3 4 1 2 2 1 4 4 3 4 3	1 2 4 3 2 1	1
1 2 3 4 3 4 1 2 2 1 4 4 3	1 2 3 4 3 4 1 2 2 3 4 3 4 3 4 3	3 2 1 4 3	0
1 2 3 4 3 4 1 2 2 3 4 3 4 3	1 2 3 4 3 4 1 2 2 3 4 4 3 4 3	3 2 1 4 3	1
1 2 3 4 3 4 1 2 2 3 4 4 3	1 2 3 4 3 4 1 2 2 3 4 1 4 3 4 3	3 2 1 4 3	1
1 2 3 4 3 4 1 2 2 3 4 1 4 3	1 2 3 4 3 4 1 2 2 3 4 1 4 1 3 4 3	3 2 1 4 3	1
1 2 3 4 3 4 1 2 2 3 4 1 4 1 3	1 2 3 4 3 4 1 2 2 3 4 1 4 1 2 3 4 3	3 2 1 4 3	1
1 2 3 4 3 4 1 2 2 3 4 1 4 1 2 3	1 2 4 1 1 1 4 3 4 3		-



CENTRO DE TECNOLGIA DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE



