

MAC122 – Princípios de Desenvolvimento de Algoritmos EP no. 2

Descrição do Problema:

O jogo do 8-Puzzle é um jogo de tabuleiro de blocos deslizáveis, que consiste em um tabuleiro quadrado com nove divisões (matriz), contendo números (ou letras) e um espaço vazio. Esse espaço vazio permite que as peças vizinhas possam “deslizar”. Ou seja, uma peça ao lado de um espaço vazio pode ser movida para essa lacuna, criando uma nova posição de lacuna. Em outras palavras, a lacuna pode ser trocada de posição com uma peça adjacente (horizontal e verticalmente). Dessa maneira pode-se mudar a configuração inicial da matriz, movimentando-se as peças. A meta do jogo é chegar a uma certa configuração final.

Ex:

2		3
1	8	4
7	6	5

a) Config. inicial

	2	3
1	8	4
7	6	5

b)

1	2	3
	8	4
7	6	5

c)

1	2	3
8		4
7	6	5

d) Config. final

O jogo consiste, portanto, em movimentar as peças partindo de uma configuração inicial (com as peças todas desordenadas), com o objetivo de chegar à configuração final. O projeto consiste em escrever um programa em linguagem C que resolva o 8-puzzle. Seu programa deve mostrar a solução do jogo, mostrando as regras aplicadas a cada passo no caminho da solução.

Estrutura geral do algoritmo

Para resolver o problema precisamos especificar a estrutura de dados para representar o estado do jogo, as regras aplicáveis e a estratégia de controle.

O estado do jogo pode ser representado por uma matriz 3x3 de caracteres onde o caractere de espaço em branco representa o “espaço vazio” do tabuleiro.

Conjunto de regras (movimentos) aplicáveis:

R1: mover ' ' para a esquerda (se a lacuna não estiver na coluna 0).

R2: mover ' ' para cima (se a lacuna não estiver na linha 0).

R3: mover ' ' para a direita (se a lacuna não estiver na coluna 2).

R4: mover ' ' para baixo (se a lacuna não estiver na linha 2).

A estratégia de controle consiste em aplicar repetidamente regras a estados do problema (matrizes) até que o estado final seja alcançado. Essa estratégia também deve “tomar conta” das regras aplicadas de modo a compor a solução para o problema, isto é, a sequência de movimentos que leva do estado inicial ao estado final.

A estratégia deve “tentar” uma regra e, se mais tarde descobrir que essa regra não era apropriada, “voltar atrás” e tentar outra regra. Essa estratégia é chamada ***Backtracking*** (Tentativa e Erro).

Suponha que as regras são selecionadas de acordo com o esquema arbitrário R1,R2,R3,R4. *Backtracking* ocorrerá nas seguintes situações:

- a) sempre que for gerado um estado anterior,
- b) sempre que se tiver aplicado um número limite de regras sem se ter chegado à solução (limite de profundidade da busca),
- c) sempre que não houver regras aplicáveis.

Data limite para Entrega: 15/11/2011.