

Estrutura de Dados e Aplicações

Trabalho

Solução de um labirinto

Nesse trabalho iremos encontrar um caminho para sair de um labirinto.

O labirinto será um tabuleiro quadriculado com $m \times n$ casas distribuídas em m linhas e n colunas. As casas livres (caminhos livres) são marcadas por 0 e as casas ocupadas (paredes do labirinto) são marcadas por 1.

Na casa $(0, 0)$, que sempre é livre, existe um robô que deve encontrar a saída do labirinto, que é a posição $(m - 1, n - 1)$, e pode se mover nas seguintes direções **norte**, **leste**, **sul** e **oeste**, uma casa por vez, desde que a casa a ser alcançada seja livre. Você deve descobrir se existe um caminho para a saída do labirinto e, se o caminho existir, imprimir a distância entre a casa $(1, 1)$ e a casa $(m - 1, n - 1)$.

Você deve usar o código do programa fornecido que possui a função para imprimir o caminho percorrido já implementada para fazer sua implementação. O código fornecido já possui o main que irá rodar o caso de teste que será cadastrado no run.codes.

Exemplo:

Labirinto (4x4):

Distância da casa inicial à casa final: 6

```
0000
1010
1010
1100
```

Tarefas

Para resolver o problema, você deve utilizar uma **fila com a implementação em lista encadeada**. Essa implementação já foi feita no código fornecido com as funções?

- **insereFila (int linha, int coluna, Fila *f);** que insere um elemento no fim da fila;
- **No *removeFila (Fila *f);** que retira um elemento do início da fila;
- **void imprimeFila (Fila *f);** que imprime a fila atual começando pelo início;

- **int resolveLabirinto (int labirinto[LIN][COL], int m, int n);** que retorna 1 e imprime o valor da distância entre a casa inicial e a casa final, caso exista um caminho até a saída, ou retorna 0, caso não exista um caminho.

O código fornecido segue o seguinte modelo:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define LIN 4
5 #define COL 4
6
7 typedef struct no No;
8 struct no{
9     int linha;
10    int coluna;
11    No *prox;
12 };
13
14 typedef struct fila Fila;
15 struct fila{
16     No *inicio;
17     No *fim;
18     int tamanho;
19 };
20
21 void insereFila (int linha, int coluna, Fila *f);
22 No *removeFila (Fila *f);
23 void imprimeFila (Fila *f);
24 int resolveLabirinto (int labirinto[LIN][COL], int m, int n)
    ;

```

O valor das macros LIN e COL estão fixados no código fornecido por 4x4, para a dimensão dos matrizes a serem passadas para a função resolveLabirinto.

Bônus: Caminho percorrido

Para validar sua função resolveLabirinto(), foi fornecida no código uma função que além da distância, ela imprime o caminho percorrido no labirinto.

Exemplo: para a entrada:

```
int labirinto4[4][4] = 0,0,0,0, 0,1,1,0, 0,1,1,0, 1,1,1,0; resolveLabirinto (labirinto4, 4, 4);
```

A saída fornecida pelo programa é:

```
(0, 0) -> (1, 0) -> (2, 0) -> (3, 0) -> (3, 1) -> (3, 2) -> (3, 3)
(0) (0) (0) (0)
(0) (1) (1) (0)
(0) (1) (1) (0)
```

(1) (1) (1) (0)

A distancia da casa inicial a final: 6 Caminho percorrido: (0, 0) -> (0, 1)
-> (0, 2) -> (0, 3) -> (1, 3) -> (2, 3) -> (3, 3)