

Curso Técnico em Informática

Laboratório de Estruturas de Dados

Trabalho Prático

Turma: 3INF
Bimestre: 02/2022

Disponibilizado em: 22/06/2022

Local de entrega: (SIGAA)

As tarefas e datas de entrega estão abaixo

Valor: 15 pontos

Prof. Anderson Grandi Pires

Objetivos:

- Implementar um programa que empregue algumas estruturas de dados estudadas na disciplina, tais como vetor dinâmico, par, fila e pilha.
- Utilizar bibliotecas disponibilizadas por terceiros, tais como a biblioteca STL da linguagem de programação C++.
- Empregar arquivos como fonte de entrada para problemas em geral.

Observações:

- A **linguagem C++ deverá ser utilizada** para implementar a atividade. Utilize as estruturas de dados da biblioteca STL que já foram apresentadas na disciplina para representar os dados do problema.
- O programa **deve funcionar** com labirintos **maiores**, desde que sejam definidos por matrizes $N \times N$ ($7 \leq N \leq 1000$) e utilizem os caracteres '#' (parede) e '.' (célula livre). A descrição dos labirintos deverá ser obtida a partir de arquivos-texto.
- Seu programa **deve apresentar** uma saída semelhante àquela apresentada no **Exemplo de execução** disponível no fim deste documento.
- Insira o nome completo de cada integrante do grupo em cada arquivo que compõe o projeto, na forma de um comentário geral (início do arquivo).
- **O programa será testado com labirintos de vários tamanhos.** A entrada e a saída sempre estarão nas células indicadas no enunciado.

TAREFAS A SEREM AVALIADAS E DATAS DE ENTREGA

Entrega dia 27/06/2022 (Valor 5 pontos)

- a) [3 pontos] Processar o arquivo "labirinto.txt" com a descrição de um labirinto (qualquer arquivo que possua um conteúdo compatível com a descrição de um labirinto apresentada na Figura 1-a deve funcionar). Um labirinto deverá ser representado por uma matriz $N \times N$, sendo $7 \leq N \leq 1000$.
- b) [2 pontos] Criar e inicializar a matriz conforme informações apresentadas na Figura 1-c.

Entrega dia 04/07/2022 (Valor 5 pontos)

- c) [3 pontos] Criar uma função para executar o passo de anotação, conforme descrito no enunciado.
- d) [2 pontos] Apresentar na tela as informações referentes às três partes da Figura 2, ou seja: Figura 2-a, Figura 2-b e Figura 2-c. A Figura 2-a deverá ser apresentada com substituindo os caracteres '.' por espaços ' ' (conforme primeira figura do exemplo de execução abaixo).

Entrega dia 11/07/2022 (Valor 5 pontos)

- e) [3 pontos] Criar uma função para implementar as funcionalidades de extração do menor caminho.
- f) [2 pontos] Apresentar na tela as informações referentes ao melhor caminho, ou seja, apresentar a Figura 3-b, caso exista um caminho. Não havendo um caminho, apresentar na tela tal informação.

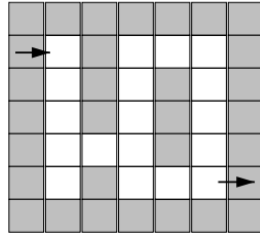
Projeto Prático – Melhor (menor) caminho em um labirinto

ETAPA 1

Dado um arquivo-texto com a descrição de um labirinto cujo tamanho deverá ser obtido dos dados do arquivo, desenvolva um programa para processar o arquivo, armazenar as informações em uma matriz com valores 0 e -1 (Figura 1-c) e apresentar na tela a representação do labirinto, ou seja, as células livres e as células ocupadas (paredes). O arquivo-texto é composto dos caracteres '.' e '#' (Figura 1-a). O caractere '.' representa uma célula livre onde um agente virtual poderá se movimentar, enquanto o caractere '#' representa uma célula bloqueada, ou seja, pode ser considerada como parte de uma parede. Um agente virtual não poderá se movimentar para uma célula bloqueada. A movimentação possível do agente será para uma célula livre adjacente à posição atual, ou seja, para cima, para baixo, para esquerda ou para direita (Figura 1-d). As figuras abaixo descrevem: (a) um exemplo do conteúdo do arquivo "labirinto.txt", (b) a representação do labirinto referente àquele arquivo, (c) a matriz a ser criada e inicializada a partir das informações do arquivo e das demais informações disponibilizadas neste documento e (d) as regras de movimentação do agente virtual (Figura 1-d).

```
# # # # # # #
# . # . . . #
# . # . # . #
# . # . # . #
# . . . # . #
# . # . . . #
# # # # # # #
```

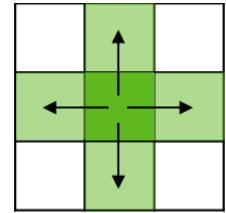
(a) labirinto.txt



(b) representação

```
-1 -1 -1 -1 -1 -1 -1
-1 0 -1 0 0 0 -1
-1 0 -1 0 -1 0 -1
-1 0 -1 0 -1 0 -1
-1 0 0 0 -1 0 -1
-1 0 -1 0 0 0 -1
-1 -1 -1 -1 -1 -1 -1
```

(c) matriz



(d) movimentação

Figura 1. Matrizes bidimensionais representando um labirinto e a regra de movimentação de um agente.

O labirinto será representado por uma matriz quadrada $L[n, n]$, onde: $L[i, j] = 0$ indica que a posição (i, j) está livre; e $L[i, j] = -1$ indica que a posição (i, j) está obstruída por uma parede. Assumimos que a entrada do labirinto está na célula $(1, 0)$ e que sua saída está na célula $(n-2, n-1)$, como indicado na Figura 1-b.

ETAPA 2

Uma vez que a primeira etapa foi concluída, o próximo passo é efetuar a etapa de **anotação**, conforme descrita abaixo:

Anotação: Para encontrar um menor caminho, primeiro anotamos na matriz L o número mínimo de passos necessários para atingir cada uma das posições do labirinto, a partir da entrada. Assim, após a fase de anotação, $L[i, j] = k$ indica que são necessários k passos para alcançar a posição (i, j) , como exemplificado na Figura 2-c. Caso uma posição qualquer da matriz permaneça com valor 0, isso significa que não existe um caminho que leve da entrada até essa posição.

Algoritmo Anotar(L)

$L[1,1] \leftarrow 1$

insira a posição (1, 1) em uma fila F vazia

enquanto F não estiver vazia **faça**

remova uma posição (i, j) da fila F

para cada posição (x, y) adjacente a (i, j) tal que $L[x, y] = 0$ **faça**

$L[x, y] \leftarrow L[i, j] + 1$

insira a posição (x, y) na fila F

```
# # # # # # #
# . # . . . #
# . # . # . #
# . # . # . #
# . . . # . #
# . # . . . #
# # # # # # #
```

(a) labirinto.txt

```
-1 -1 -1 -1 -1 -1 -1
-1 0 -1 0 0 0 -1
-1 0 -1 0 -1 0 -1
-1 0 -1 0 -1 0 -1
-1 0 0 0 -1 0 -1
-1 0 -1 0 0 0 -1
-1 -1 -1 -1 -1 -1 -1
```

(b) representação

```
-1 -1 -1 -1 -1 -1 -1
-1 1 -1 9 10 11 -1
-1 2 -1 8 -1 12 -1
-1 3 -1 7 -1 11 -1
-1 4 5 6 -1 10 -1
-1 5 -1 7 8 9 -1
-1 -1 -1 -1 -1 -1 -1
```

(c) anotação

Figura 2. Representações de um labirinto em (a) e (b). Resultado da etapa de anotação em (c).

ETAPA 3

A última etapa compreende na extração do menor caminho (caso exista) para ir da entrada até a saída do labirinto (Figura 3-b). O algoritmo para efetuar esta etapa está descrito a seguir:

Algoritmo Extrair(L)

se $L[n-2, n-1] = 0$ **então** não existe um caminho, **pare** a execução da função
insira a posição $(n-1, n-1)$ em uma pilha P vazia
enquanto a posição $(1, 1)$ não estiver no topo da pilha P **faça**
 seja (i, j) a posição existente no topo da pilha
 encontre uma posição (x, y) adjacente a (i, j) tal que $L[x, y] = L[i, j] - 1$
 insira a posição (x, y) na pilha P
mostre o caminho armazenado na pilha P

```
# # # # # # #
# . # . . . #
# . # . # . #
# . # . # . #
# . . . # . #
# . # . . . #
# # # # # # #
```

(a) exemplo de labirinto

```
# # # # # # #
# X #       #
# X #       #
# X #       #
# X X X #    #
#   # X X X #
# # # # # # #
```

(b) melhor caminho

Figura 3. Conteúdo do arquivo de entrada com a representação de um labirinto (a) e um melhor caminho identificado como resultado da etapa de extração (b).

EXEMPLO DE EXECUÇÃO

(Neste exemplo **NÃO** estão **TODAS** as figuras solicitadas)

```
# # # # # # #
#   #       #
#   #       #
#   #       #
#   #       #
#   #       #
# # # # # # #
```

Labirinto

```
# # # # # # #
# 1 # 9 10 11 #
# 2 # 8 # 12 #
# 3 # 7 # 11 #
# 4 5 6 # 10 #
# 5 # 7 8 9 #
# # # # # # #
```

Etapa de anotação

```
# # # # # # #
# x #       #
# x #       #
# x #       #
# x x x #    #
#   # x x x #
# # # # # # #
```

Um menor caminho