

Labirinto

Programação de Computadores 1

Prof. Daniel Saad Nogueira Nunes



1 Contextualização

Um problema muito comum em Ciência da Computação é o problema da busca de caminhos (*pathfinding*), que consiste em encontrar um caminho entre uma origem e um destino de um grafo.

Neste projeto, estamos interessados encontrar o caminho do ponto de partida até a saída do labirinto. O labirinto é um *grid* bidimensional. Este caminho pode ser obtido por meio de uma técnica recursiva, denominada busca em profundidade.

1.1 Especificação

O seu programa deverá receber via parâmetro de linha de comando os caminhos para dois arquivos texto: um arquivo de entrada e outro de saída.

O arquivo de entrada especifica o labirinto L . Este arquivo texto é descrito da seguinte maneira:

- A primeira linha do arquivo possui dois inteiros N , e M , ($10 \leq N, M \leq 10^5$), separados por espaço.
- As próximas N linhas, contém cada uma M caracteres, que podem assumir o valor ‘ ’ (espaço em branco) ou ‘#’, representando uma parede.

A entrada do labirinto sempre se encontra na célula $L[0][1]$ e a saída sempre se encontra na célula $L[N - 1][M - 2]$, ou seja, é garantido que essas células armazenam um espaço em branco.

Deve-se encontrar o caminho até a saída, marcando este caminho com o símbolo ‘@’, e escrever o labirinto resolvido no arquivo texto de saída. Os movimentos permitidos são aqueles nas 4 direções (cima, baixo, esquerda, direita), isto é, exclui-se as diagonais das possibilidades de movimento. Logicamente, não é permitido atravessar uma parede.

Observação: devido ao tamanho máximo que o labirinto pode assumir, ele deve ser armazenado em uma região de memória alocada dinamicamente.

1.2 Exemplo

Arquivo de Entrada

```
19 23
# #####
#      #
### # # ##### # #
# # # #      # # #
# # # # ##### # # #
# #   #   #   #   #
# ##### ### # # # # ###
#     #   # # #   #   #
##### # # # # ##### #
#     # # # #      # #
# ##### # ##### # #
```

```

# #      # #      # # #
# # ##### # ### # ###
# # #      # #      #
# ### ##### # ### #
#  #      #  # #
# # ##### #
# #      #
##### #

```

Arquivo de Saída

```

#@#####
#@@@#@@@      # #
###@#@#@##### # #
# #@#@#@#@#@#@#@#@#@# # #
# #@#@#@#####@# # #
# #@@@#      #@@@#@# #
# ##### ### #@#@#@# ###
#      # # #@#@#@# #
##### # # # #@##### #
#      # # # #@@@@@@@@# #
# ##### # #####@# #
# #      # #      #@# #
# # ##### # ### #@###
# # #      # #      #@@@#
# ### ##### # ###@#
#  #      #  #@#
# # #####@#
# #      @#
#####@#

```

1.3 Modularização

Para realizar esta tarefa, deverão ser criados os módulos de:

- Leitura: lê do arquivo de entrada o labirinto.
- Processamento: encontra a solução do labirinto.
- Escrita: imprime a solução
- Tratamento de erros: captura os possíveis erros na abertura de arquivos, alocação dinâmica de estruturas, etc, e reporta ao usuário, encerrando o programa graciosamente.

Além dos módulos obrigatórios, outros módulos podem ser criados a critério do programador.

1.4 Construção do sistema

Um `Makefile` deverá ser produzido para a compilação dos códigos-fontes no executável e deverá ser distribuído junto ao código.

1.5 Documentação

O código deve ser bem documentado, com presença de comentários explicando os trechos mais complexos do código. Além disso, um arquivo `README` deve ser providenciado com a devida identificação do autor descrevendo o projeto e instruindo como o código deve ser compilado através da ferramenta `make`.

2 Critérios de correção

Para validação da correção do algoritmo, testes automatizados serão realizados, então é **crucial** que a saída esteja conforme o especificado.

Além da correção do algoritmo, também serão avaliados a documentação e o `Makefile`.

Serão descontados pontos dos códigos que não possuírem indentação ou documentação.

2.1 Ambiente de Correção

Para a correção dos projetos, será utilizada uma máquina de 64-bits com sistema operacional GNU/LINUX e compilador GCC 10.2.0, logo é imprescindível que o sistema seja capaz de ser compilado e executado nesta configuração.

3 Considerações

- Considere utilizar ferramentas de depuração, como o GDB e o Valgrind para encontrar erros.
- Este trabalho deve ser feito **individualmente**.
- O trabalho que não compilar não será avaliado.
- Os trabalhos que incidirem em plágio serão avaliados automaticamente com nota 0 para os envolvidos. Medidas disciplinares também serão tomadas.
- O trabalho deverá ser entregue conforme data estabelecida na no ambiente de aprendizagem virtual da disciplina.