

Labirinto

Programação de Computadores I

Ciência da Computação

Prof. Daniel Saad Nogueira Nunes



**INSTITUTO
FEDERAL**
Brasília

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 em um *grid* bidimensional.

Neste projeto, estamos interessados em dado um labirinto em um *grid* bidimensional e um ponto de partida, encontrar um caminho do ponto de partida até a saída do labirinto. 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 , ($10 \leq N, M \leq 100$), 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.

Dada o labirinto, deve-se encontrar o caminho até a saída, marcando este caminho com o símbolo ‘@’, e escrever o labirinto resolvido em um arquivo texto de saída. Os movimentos permitidos são aqueles nas 4 direções (cima, baixo, esquerda, direita), excluindo as diagonais. Não é permitido atravessar uma parede.

É importante que o labirinto seja armazenado em uma região de memória alocada dinamicamente.

1.2 Exemplo

Arquivo de Entrada

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

Arquivo de Saída

1.3 Modularização

- Leitura do arquivo texto de entrada.
- Processamento do labirinto.
- Escrita do arquivo texto de saída.

2

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

- 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.