

Trabalho Prático



Escreva um programa que solucione o seguinte problema: um rato está em um labirinto e precisa encontrar o caminho até um pedaço de queijo. Seu programa deve ler um arquivo de entrada (que contém o labirinto, a posição do rato, a posição do queijo) e escrever o caminho no arquivo de resultado. Para encontrar e armazenar o caminho, use uma pilha (para armazenar as tentativas) e uma lista (para armazenar os passos do caminho). Este enunciado vem acompanhado de um programa para geração aleatória de arquivos de entrada. Tal arquivo tem o seguinte formato:

- **Linha 1:** Dois inteiros positivos, separados por espaço, que determinam o tamanho do labirinto, em números de linhas e colunas.
- **Linha 2:** Posição do rato, representada por dois inteiros, que correspondem a linha e coluna.
- **Linha 3:** Posição do queijo, representada por dois inteiros, que correspondem a linha e coluna.
- **Linhas seguintes:** Labirinto, representado por uma matriz, sem espaços separando as células dessa estrutura. Células sem obstáculo contêm 0, células com obstáculo contêm 1, as posições do rato e do queijo contêm *.

Após ler o arquivo de entrada e montar a matriz na classe Labirinto, seu programa deve executar o algoritmo principal: empilhar a posição inicial do rato; executar uma repetição que desempilha uma posição e empilha os vizinhos sem obstáculos (superior, esquerda, inferior, direita); terminar a repetição quando desempilhar o destino ou esvaziar a pilha. Em cada iteração, inserir em uma lista as tentativas de movimento, para evitar repetição infinita e para recuperar o caminho encontrado. O esquema abaixo mostra os passos do algoritmo para o exemplo anterior de arquivo de entrada.

Arquivo de entrada	
6 6	
1 1	
4 4	
111111	
1*0101	
100011	
101001	
1010*1	
111111	

Labirinto (R = rato, Q = queijo)						
	0	1	2	3	4	5
0	1	1	1	1	1	1
1	1	R	0	1	0	1
2	1	0	0	0	1	1
3	1	0	1	0	0	1
4	1	0	1	0	Q	1
5	1	1	1	1	1	1

Por fim, seu programa deve escrever o caminho em um arquivo; no exemplo aqui mostrado, o programa escreve o seguinte arquivo: “(1,1) (2,1) (2,2) (3,2) (3,3) (4,3) (4,4)”. Para armazenar este caminho, use uma lista encadeada. Importante: No exemplo aqui mostrado, o caminho foi encontrado de maneira direta, sem nenhum retrocesso. Em outros exemplos, o algoritmo pode tentar um caminho sem sucesso; isso acontece quando, em alguma posição, não existirem vizinhos novos e sem obstáculos para empilhar. Neste caso, é preciso remover a última posição inserida na lista e continuar o algoritmo, desempilhando a próxima posição.

Iteração	Desempilhar	Empilhar	Pilha (da base ao topo)
1		(1,1)	(1,1)
2	(1,1)	(1,2) (2,1)	(1,2) (2,1)
3	(2,1)	(2,2)	(1,2) (2,2)
4	(2,2)	(1,2) (3,2)	(1,2) (1,2) (3,2)
5	(3,2)	(3,3)	(1,2) (1,2) (3,3)
6	(3,3)	(3,4) (4,3)	(1,2) (1,2) (3,4) (4,3)
7	(4,3)	(4,4)	(1,2) (1,2) (3,4) (4,4)
8	(4,4)		(1,2) (1,2) (3,4)

Entrega: O trabalho será entregue pelo Canvas até 23:59h de 05/06/2022 e apresentado ao professor na aula seguinte. Insira os códigos de seu programa em um diretório cujo nome seja “TrabalhoPratico-Nome1-Nome2” (em duplas), compacte e submeta o arquivo compactado. Os alunos que não enviarem o trabalho até a data definida ou que não participarem da apresentação

ao professor ganharão nota zero. Importante mencionar também que trabalhos copiados ganharão nota zero, não importando quem forneceu ou recebeu a “cola”.

Critérios de avaliação: Programe com capricho, usando as estruturas adequadas e comentando o código. Use também mensagens claras, para requerer informação do usuário e para lhe responder. Use nomenclatura sugestiva para as variáveis e procure tratar problemas de execução. A correção do trabalho observará os critérios abaixo:

- Código com qualidade (organizado e comentado): 1 pontos;
- Classe Labirinto corretamente implementada: 1 pontos;
- Algoritmo principal corretamente implementado, conforme descrito aqui: 7 pontos;
- Leitura de arquivo no formato correto: 0,5 ponto;
- Escrita de arquivo no formato correto: 0,5 ponto;