

Modelagem Computacional em Grafos

SCC0216

Disponibilizado em: 24-04-2023

Código Turma A (terça): **6NGE**

Submissões até: 21-05-2023

Código Turma B (segunda): **1KNW**

Trabalho 2 - Pacman

1. Instruções

Este trabalho deve ser feito em grupos de 2 ou 3 alunos. Apenas um aluno do grupo deve fazer a submissão no RunCodes. Os grupos devem se registrar preenchendo este formulário (<https://forms.gle/j7y3Sv9fu5KEHZMy9>), indicando a composição do grupo e qual dos membros vai fazer a submissão (o primeiro na lista).

2. Introdução

Pacman é um dos jogos mais populares da história, lançado em 1980 pela empresa japonesa Namco (<https://pt.wikipedia.org/wiki/Pac-Man>). O objetivo do jogo é controlar um personagem que se movimenta em um labirinto enquanto tenta evitar os fantasmas que também se movimentam pelo cenário.

Em termos de computação, é um exemplo clássico de um jogo que envolve algoritmos de busca. Os fantasmas, por exemplo, usam algoritmos de busca para determinar a melhor rota para alcançar o Pacman. O desafio para o jogador (o Pacman) é evitá-los. Em algumas situações, o Pacman adquire invencibilidade e ao invés de perder uma vida quando encosta em um fantasma ele consegue derrotá-lo e ganhar o jogo, como ilustrado na figura a seguir:



Figura 1: Exemplo do jogo pacman

Neste trabalho, você é o Pacman depois de adquirir esse poder e consegue derrotar um fantasma apenas chegando até ele. Porém, para que isso aconteça, você precisa, a partir de uma lista de movimentos do fantasma, chegar até ele fazendo o menor número de movimentos possíveis, evitando os obstáculos no seu caminho!

3. Entrada

- N, indicando que o cenário é um tabuleiro $N \times N$ (entrada padrão)
- As próximas N linhas contém N números cada: cada número corresponde a uma célula do tabuleiro: se $Tab(i,j) = 0$, o espaço é vazio, se $Tab(i,j) = 1$, indica um obstáculo (apenas o fantasma pode adentrar uma célula com um obstáculo, o pacman não)
- Os inteiros i e j que identificam a célula (posição no tabuleiro) $Tab(i,j)$ na qual o pacman se encontra inicialmente
- Os inteiros i e j que identificam a célula (posição no tabuleiro) $Tab(i,j)$ na qual o fantasma se encontra inicialmente
- O número M de movimentos do fantasma (entrada padrão)
- Uma lista dos M movimentos do fantasma (entrada padrão)

A partir de uma célula qualquer, o personagem pode se movimentar em quatro direções (cima, baixo, direita e esquerda), ele não pode sair das bordas do tabuleiro nem ultrapassar obstáculos.

O fantasma pode fazer um número n de movimentos pré-definidos que serão lidos da entrada padrão :

- 'L': movimento para esquerda
- 'R': movimento para a direita
- 'U': movimento para cima
- 'D': movimento para baixo

O fantasma pode atravessar células do tabuleiro com obstáculo, mas não pode sair do tabuleiro também. Obs: Depois que acabarem os movimentos do fantasma, ele ficará parado na última posição em que esteve, sendo obstáculo ou não.

Obs: A leitura de caracteres em sequência pode causar confusão em C, uma recomendação é usar a máscara:

```
scanf(" %c", &character);
```

Exemplo (ver item 5 abaixo):

A partir da leitura da entrada padrão, o valor de N informado foi 3. Você deverá, então, receber o tabuleiro, que é uma matriz $N \times N$:

```
0 -- 0 -- 0
|    |    |
0 -- 0 -- 0
|    |    |
0 -- 0 -- 0
```

Figura 2: Representação em matriz do tabuleiro

Suponha que o Pacman(**P**) foi posicionado inicialmente na célula (1,0), enquanto o fantasma(**Q**) foi posicionado na célula (2,2).

```
0 -- 0 -- 0
|    |    |
P -- 0 -- 0
|    |    |
0 -- 0 -- Q
```

Figura 3: Tabuleiro com o pacman e o fantasma posicionados

Para manter um padrão de movimentação, sempre que houver opção entre os movimentos na horizontal e na vertical, o Pacman prioriza a direção horizontal para se aproximar do fantasma. Assim, o Pacman então realiza sua primeira movimentação para a direita, indo para a posição (1,1):

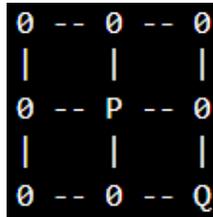


Figura 4: Tabuleiro após primeiro passo do Pacman

Depois da primeira movimentação do Pacman, é realizada a primeira movimentação do fantasma, nesse caso, é lido da entrada um 'U'. Teremos, então:

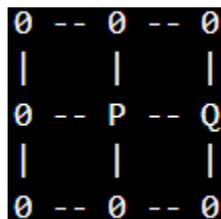


Figura 5: Tabuleiro após movimentação do fantasma

Agora, em seu segundo passo, o Pacman se move para a direita e acaba no mesmo nó que o fantasma.

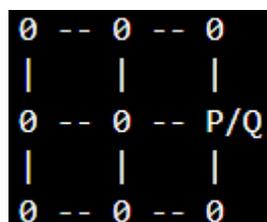


Figura 6: Tabuleiro após segundo passo do Pacman

O Pacman conseguiu, então, atingir o fantasma em 2 passos.

4. Saída

A saída esperada do seu programa é:

- O número mínimo de passos realizados **pelo Pacman** até chegar no fantasma.
- As movimentações realizadas até que isso ocorra (para direita, esquerda, cima e baixo).
- OBS: Independente do número de movimentações fornecidas do fantasma, **assim que o Pacman e o le estiverem na mesma posição**, a saída deverá ser fornecida e o programa finalizado.
- Caso não seja possível chegar no fantasma com o pacman em qualquer número de movimentos, exiba:
"Não foi possível achar um caminho" (sem as aspas)

5. Exemplos de entrada e saída

Entrada: 3 0 0 0 0 0 0 0 0 0 1 0 2 2 1 U	Saída: Número de passos: 2 Movimentos para cima: 0 Movimentos para baixo: 0 Movimentos para esquerda: 0 Movimentos para direita: 2
---	--

Entrada: 4 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 3 3 2 UL	Saída: Número de passos: 4 Movimentos para cima: 0 Movimentos para baixo: 2 Movimentos para esquerda: 0 Movimentos para direita: 2
---	--

Entrada: 4 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 3 3 0	Saída: Não foi possível achar um caminho
---	--

Bom Trabalho :)