

PTQPRS

Projeto do Robô

Restrição de Complexidade Assintótica na Solução: NÃO

Tempo máximo: 60 segundos

Você projetou um robô que percorre os cômodos de sua residência e, neste processo, ele faz uma limpeza do piso. Este robô se move para frente e pode rotacionar em ângulos de 90° , portanto, se move em 4 direções (p/ cima, p/ baixo, p/ esquerda, p/ direita). Além disso, ele se move apenas entre os cômodos onde a porta de acesso está aberta. Inicialmente, uma porta fechada não é um problema, pois sua casa é inteligente e periodicamente ela envia o mapa atualizado da casa ao robô, ou seja, um mapa marcando os cômodos abertos e fechados. Deste modo seu robô é capaz de identificar se ele consegue ou não entrar no cômodo para limpar. Hoje faz 30 dias que seu robô está ajudando na limpeza da casa. Então você resolveu perguntar aos seus familiares se o robô está funcionando corretamente. Sua mãe lhe respondeu: “Ele ajuda bastante. Entretanto, ele poderia automaticamente identificar que vai ficar sem bateria e, neste instante, poderia voltar ao ponto de carga. Isso evita que as pessoas pisem nele”. Você pensou que isso é possível pois, com um sensor, seu robô consegue saber que a bateria está fraca. Então ele pode solicitar a casa o mapa atualizado e, com isso, ele pode identificar uma rota que o leva a base de carga.

Então agora temos que atualizar o sistema do robô. Para isso projete um algoritmo que recebe como entra: 1) o mapa atualizado da casa; 2) o atual ponto onde o robô se encontra; 3) o ponto onde se encontra a base de carga; e gera como saída se ele consegue ou não chegar no ponto de carga.

Entrada

Cada entrada é formada por um único caso de teste. Na primeira linha de cada caso há dois números, representando as dimensões do mapa atualizado: 1) i Linhas – quantidade de linhas existentes no mapa; 2) j Colunas – quantidade de colunas no mapa. As próximas i Linhas descrevem o mapa, que é formado por quatro caracteres diferentes: 1) espaço – representando uma área livre ou uma porta aberta; 2) # (hashtag) – representando um obstáculo intransponível, seja uma parede, seja uma porta fechada; 3) S (letra em uppercase) – representando o atual local do robô; 4) D (letra em uppercase) – representando o ponto de carga do robô. O mapa tem dimensão mínima de 3 linhas por 4 colunas, formando um quadrado ou um retângulo. Além disso, a primeira linha, a última linha, a primeira coluna e a última coluna, são paredes ou portas fechadas. Considere que há apenas um robô e um ponto de carga.

Saída

Para cada caso de teste, seu programa deve imprimir: “fica no mesmo lugar” (sem aspas) caso o robô não consiga voltar ao ponto de carga, ou “vai ao ponto de carga” (sem aspas) caso tenha

um caminho livre e possível para voltar ao ponto de carga.

Restrições da Implementação C

- Seu código não pode usar rotinas de biblioteca para manipular/alterar os dados.
 - Apenas as rotinas de biblioteca p/ interação com o teclado/tela (ex., printf, scanf, gets, ...) e as rotinas relacionadas à alocação de memória (ex. malloc, calloc, ...) são permitidas.
- Seu programa deve usar alguma implementação de árvore e/ou grafo (use apenas as que estudamos).
 - Não utilize Vetores/matrizes como estrutura de dados, no lugar, utilize alguma implementação de lista encadeada.
 - Seu programa DEVE usar o algoritmo BFS **ou** DFS.
- Seu programa NÃO deve utilizar variáveis globais.
- GARANTA que não haverá desperdício de memória, seja por alocação extra, seja por não liberar a memória.
- Seu programa deve utilizar no mínimo três funções e/ou procedimentos, criados por você.
- Seu programa deve usar APENAS algoritmos que estudamos e foram apresentados em aula

Example

Input:

```
6 27
#####
#S  ##      #
#  ##      ## # #
#  ##      ## # #
#          ## # D#
#####
```

Output:

vai ao ponto de carga

Example

Input:

```
7 30
#####
#      #####      #####
##### ###      ## # # # #
#S      ### ## ## ##### #
##### #####      #  # #
#  ##      #####D #####
#####
```

Output:

fica no mesmo lugar

