Nome: ____________________

RA: __________ Assinatura: __________

Observações:

- Leia com cuidado a pergunta antes de começar. Você pode solicitar esclarecimentos durante o teste.
- Explique e justifique a sua resposta.
- Você também pode obter pontos por uma resposta parcial.
- É proibido qualquer tipo de consulta bibliográfica.
- O teste é individual, qualquer detecção de fraude implicará em zerar a nota da disciplina.

Questão 1. Você recebe uma grade $m \times n$ composta por células livres e células bloqueadas. Um robô inicia na célula $(1, 1)$ e deseja chegar à célula (m, n). A cada movimento, o robô pode ir apenas para a direita ou para baixo, e só pode se deslocar por células livres. Desejamos determinar o menor número de movimentos necessários para chegar do início ao fim. Caso não exista nenhum caminho possível, sua resposta deve ser ∞.

(a) **(5 pontos)** Projete um algoritmo de programação dinâmica para solucionar o problema. Mostre a existência de subestrutura ótima: você deve definir a recorrência e justificar sua correção.

(b) **(2 pontos)** Escreva o pseudocódigo do seu algoritmo.

(c) **(3 pontos)** Qual a complexidade do seu algoritmo? Justifique.

Resposta

(a) Denote por $T[i, j]$ o menor número de movimentos para chegar da célula (i, j) à (m, n), então:

$$T[i,j] = \begin{cases} 0, & \text{se } (i,j) = (m,n), \\ \infty & \text{se a célula } (i,j) \text{ é bloqueada,} \\ 1 + T[i+1,j], & \text{se } i < m, j = n, \\ 1 + T[i,j+1], & \text{se } i = m, j < n, \\ 1 + \min\{T[i+1,j], T[i,j+1]\}, & \text{se } i < m, j < n. \end{cases}$$

Usemos indução reversa em para provar a recorrência:

- **Caso básico.** $i = m$ e $j = n$, já estamos no destino. Portanto, o menor número de movimentos é 0.
- **Passo.** $i < m$ ou $j < n$. Analisemos cada caso da recorrência:
  - Se (i, j) é bloqueada, então não é possível ocupar essa célula. Logo, não existe caminho que passe por ela e o valor deve ser ∞.
  - Se $j = n$, estamos na borda direita da grade e não podemos nos mover para a direita. A única opção é se movimentar para baixo e seguir pelo melhor caminho a partir de $(i + 1, j)$. Assim, o menor número de movimentos é $1 + T[i + 1, j]$, que, pela hipótese de indução, sabemos está correto.
  - Se $i = m$, estamos na borda inferior da grade e só podemos nos mover para a direita. Portanto, de forma análoga ao caso anterior, o menor número de movimentos é $1 + T[i, j + 1]$.
  - Senão, $i < m$ e $j < n$ e podemos ir tanto para baixo quanto para a direita. Um caminho ótimo a partir de (i, j) consiste em um movimento inicial seguido do melhor caminho a partir $(i + 1, j)$ ou de $(i, j + 1)$. Pela hipótese de indução, ambos os valores já foram corretamente calculados, e basta escolher o menor entre eles. Assim, o valor ótimo é $1 + \min\{T[i + 1, j], T[i, j + 1]\}$.

  Em todos os casos, mostramos que a recorrência atribui corretamente o menor número de movimentos.

(b) Podemos escrever o algoritmo na versão bottom-up como segue:

Algoritmo 1: Menor-Caminho(m, n)

$T[m, n] \leftarrow 0$
para $i \leftarrow m$ **até** 1
    para $j \leftarrow n$ **até** 1
        se (i, j) *está bloqueada*
            $T[i, j] \leftarrow \infty$
        senão
            se $i < m$ *e* $j = n$
                $T[i, j] \leftarrow 1 + T[i + 1, j]$
            se $i = m$ *e* $j < n$
                $T[i, j] \leftarrow 1 + T[i, j + 1]$
            se $i < m$ *e* $j < n$
                $T[i, j] \leftarrow 1 + \min\{T[i + 1, j], T[i, j + 1]\}$
devolva $T[1, 1]$

(c) As linhas internas do algoritmo (linhas 4 a 12) executam apenas operações de comparação, acesso, soma e atribuição, cada uma custando $\Theta(1)$. Esse bloco é executado para cada par (i, j) da grade, ou seja:

- O laço externo (linha 2) executa m vezes.
- O laço interno (linha 3) executa n vezes.

Assim, o custo total do laço duplo é: $\Theta(m \cdot n)$. As linhas fora dos laços (1 e 17) têm custo $\Theta(1)$, o que não altera a ordem de complexidade total. Portanto, a complexidade total do algoritmo é $\Theta(m \cdot n)$.