

# Tutorial: Caminho de Menor Soma

## 1 Solução do Problema

O problema consiste em encontrar o **menor custo** necessário para percorrer uma grade (*matriz*) de inteiros positivos. A grade possui  $n$  linhas e  $m$  colunas, e cada célula contém um número positivo que representa o custo de passar por ela.

O objetivo é sair da célula **superior esquerda** da grade e chegar à célula **inferior direita**, movendo-se apenas para a **direita** ou para **baixo**. A soma dos valores das células visitadas deve ser a menor possível.

Este problema pode ser resolvido de forma eficiente utilizando **programação dinâmica**.

### 1.1 Definição do Subproblema

Seja uma matriz de custos  $C$  de tamanho  $n \times m$ . Definimos:

$dp[i][j]$  = o menor custo para chegar à célula  $(i, j)$  a partir de  $(0, 0)$ .

Ou seja,  $dp[i][j]$  representa o custo mínimo para alcançar exatamente a posição  $(i, j)$ .

### 1.2 Função de Transição

Como só é possível mover-se para a **direita** ou para **baixo**, a célula  $(i, j)$  só pode ser alcançada a partir de:

- $(i - 1, j)$ : movimento vindo de **cima**;
- $(i, j - 1)$ : movimento vindo da **esquerda**.

Assim, o custo mínimo para chegar a  $(i, j)$  é dado por:

$$dp[i][j] = C[i][j] + \min(dp[i - 1][j], dp[i][j - 1]).$$

### 1.3 Casos Base

- Primeira célula:

$$dp[0][0] = C[0][0]$$

- Primeira linha (somente movimentos para a direita):

$$dp[0][j] = dp[0][j - 1] + C[0][j]$$

- Primeira coluna (somente movimentos para baixo):

$$dp[i][0] = dp[i - 1][0] + C[i][0]$$

## 1.4 Construção da Tabela

Após inicializar os casos base, preencheremos o restante da matriz  $dp$  utilizando:

$$dp[i][j] = C[i][j] + \min(dp[i-1][j], dp[i][j-1]).$$

Ao término do processo, o valor da última posição:

$$dp[n-1][m-1]$$

representa o **menor custo possível** para ir da célula inicial até a célula final.