HyperionDev

Data Structures – 2D Lists

Model-Answer Approach

Visit our website

Auto-graded task 1

The provided solution tackles the Minesweeper problem, aiming to determine the number of neighbouring mines for each cell in the grid. It first verifies the grid's validity, ensuring it's not empty and has consistent dimensions. Then, it iterates through each cell, checking its neighbouring cells for mines in eight directions: east, west, north, south, and diagonals. For each mine found, it increments a counter. After examining all neighbours, it updates the current cell's value with the total count of neighbouring mines. The solution handles potential pitfalls by raising a `ValueError` for invalid grids, such as empty or irregular dimensions. Additionally, it includes a `print_grid` function to display the grid in a formatted manner, improving readability. Overall, this approach offers a systematic and robust solution to the Minesweeper problem, ensuring accurate mine counts while addressing potential errors.