

# Approach and In-Depth Explanation

## Intuition

The problem requires finding the minimum initial health required for the knight to reach the princess' cell without dying. The knight can move only down or right. We need to ensure that at each cell, the knight has enough health to reach the princess.

## Approach

We can use dynamic programming to solve this problem. We create a DP table with dimensions  $(m+1) \times (n+1)$ , where  $m$  and  $n$  are the dimensions of the grid. We initialize all values in the DP table to `INT_MAX`. Then, we set the cells to the bottom and right of the princess' cell with a value of 1, indicating that the knight can survive with at least 1 health point on these cells.

Next, we iterate over the DP table from bottom-right to top-left. For each cell, we calculate the minimum health required to reach the princess' cell from that cell. We do this by considering the minimum health required to reach the next cells (down and right) and subtracting the points at the current cell.

Finally, we return the minimum health required to reach the princess' cell, which is stored in the top-left cell of the DP table.

## Complexity

- Time complexity:  $O(m*n)$ , where  $m$  and  $n$  are the dimensions of the grid.
- Space complexity:  $O(m*n)$ , the space used by the DP table.