

```

def maxMoves(grid):
    def dfs(row, col):
        if row < 0 or row >= m or col >= n:
            return 0
        moves = 0
        current_val = grid[row][col]

        # Try all possible moves
        for newRow, newCol in [(row - 1, col + 1), (row, col + 1), (row +
1, col + 1)]:
            if 0 <= newRow < m and 0 <= newCol < n and
grid[newRow][newCol] > current_val:
                moves = max(moves, 1 + dfs(newRow, newCol))
        return moves

    m, n = len(grid), len(grid[0])
    max_moves = 0

    # Iterate through each cell in the first column and find the maximum
moves
    for i in range(m):
        max_moves = max(max_moves, dfs(i, 0))

    return max_moves

# Example
grid1 = [[2,4,3,5],[5,4,9,3],[3,4,2,11],[10,9,13,15]]
grid2 = [[3,2,4],[2,1,9],[1,1,7]]
print(maxMoves(grid1))
print(maxMoves(grid2))

```