

## Longest Increasing Path in a Matrix

Given an  $m \times n$  integers matrix, return *the length of the longest increasing path in matrix*.

From each cell, you can either move in four directions: left, right, up, or down. You **may not** move **diagonally** or move **outside the boundary** (i.e., wrap-around is not allowed).

**Example 1:**

9	9	4
6	6	8
2	1	1

Diagram illustrating the longest increasing path in the matrix. The path is highlighted by arrows: 1 (bottom row, middle) → 2 (bottom row, left) → 6 (middle row, left) → 9 (top row, left).

**Input:** matrix = [[9,9,4],[6,6,8],[2,1,1]]

**Output:** 4

**Explanation:** The longest increasing path is [1, 2, 6, 9].

**Example 2:**

3	4	5
3	2	6
2	2	1

Diagram illustrating the longest increasing path in the matrix. The path is highlighted by arrows: 3 (top row, left) → 4 (top row, middle) → 5 (top row, right) → 6 (middle row, right).

**Input:** matrix = [[3,4,5],[3,2,6],[2,2,1]]

**Output:** 4

**Explanation:** The longest increasing path is [3, 4, 5, 6]. Moving diagonally is not allowed.

**Example 3:****Input:** matrix = [[1]]**Output:** 1**Constraints:**

- $m == \text{matrix.length}$
- $n == \text{matrix}[i].\text{length}$
- $1 \leq m, n \leq 200$
- $0 \leq \text{matrix}[i][j] \leq 2^{31} - 1$