**Rapid Waters**

The waters are rising! You must swim from the top left cell to the bottom right cell given the following constraints. Rain will begin to fall. At time **T**, the depth of the water is **T**. You can swim from a cell to any adjacent cell (left, right, up, down) only if the adjacent cell is less than or equal to the current depth. You can swim infinite distances in zero time as long as you’re within the boundaries of the map.

**Input:** The first line of input contains **C**, the number of test cases. The first line of each test case will have **N**, the rows and columns of the grid. The next **N** lines will have **N** space-separated integers that will represent the depth that must be reached before swimming on that cell.

**Output:** You will first output “CASE #(case number): “ followed by the least amount of time you must take before reaching the bottom right cell.

**Example Input:**

2

2

0 2

1 3

5

0 1 2 3 4

24 23 22 21 5

12 13 14 15 16

11 17 18 19 20

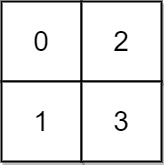
10 9 8 7 6

**Example Output:**

CASE #1: 3

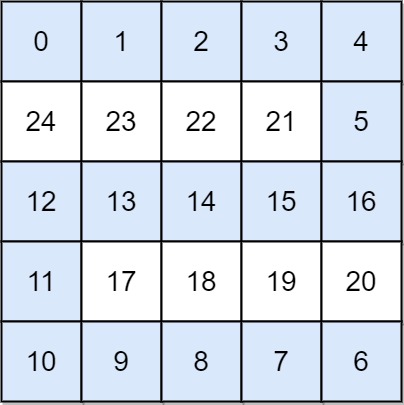
CASE #2: 16

**Explanation:** For the first test case, the map will appear as such:



At time 0, you cannot move anywhere. You cannot reach (1, 1) until time 3. By that point, you can move anywhere on the map.

For the second test case, the map and the given path will look like this:



The minimum time you can take is 16, since that is the longest time along the path.