Problem 2387: Median of a Row Wise Sorted Matrix

Problem Information

Difficulty: Medium
Acceptance Rate: 70.17%
Paid Only: Yes
Tags: Array, Binary Search, Matrix

Problem Description

Given an `m x n` matrix `grid` containing an **odd** number of integers where each row is sorted in **non-decreasing** order, return _the**median** of the matrix_.

You must solve the problem in less than `O(m * n)` time complexity.

Example 1:

Input: grid = [[1,1,2],[2,3,3],[1,3,4]] **Output:** 2 **Explanation:** The elements of the matrix in sorted order are 1,1,1,2,_2_ ,3,3,3,4. The median is 2.

Example 2:

Input: grid = [[1,1,3,3,4]] **Output:** 3 **Explanation:** The elements of the matrix in sorted order are 1,1,_3_ ,3,4. The median is 3.

Constraints:

* `m == grid.length` * `n == grid[i].length` * `1 <= m, n <= 500` * `m` and `n` are both odd. * `1 <= grid[i][j] <= 106` * `grid[i]` is sorted in non-decreasing order.

Code Snippets

C++:

```
class Solution {
public:
int matrixMedian(vector<vector<int>>& grid) {

}
};
```

Java:

```
class Solution {
public int matrixMedian(int[][] grid) {

}
}
```

Python3:

```
class Solution:
def matrixMedian(self, grid: List[List[int]]) -> int:
```