Problem 74: Search a 2D Matrix

Problem Information

Difficulty: Medium
Acceptance Rate: 53.11%
Paid Only: No
Tags: Array, Binary Search, Matrix

Problem Description

You are given an `m x n` integer matrix `matrix` with the following two properties:

* Each row is sorted in non-decreasing order. * The first integer of each row is greater than the last integer of the previous row.

Given an integer `target`, return `true` _if_ `target` _is in_ `matrix` _or_ `false` _otherwise_.

You must write a solution in `O(log(m * n))` time complexity.

Example 1:

Input: matrix = [[1,3,5,7],[10,11,16,20],[23,30,34,60]], target = 3 **Output:** true

Example 2:

Input: matrix = [[1,3,5,7],[10,11,16,20],[23,30,34,60]], target = 13 **Output:** false

Constraints:

* `m == matrix.length` * `n == matrix[i].length` * `1 <= m, n <= 100` * `-104 <= matrix[i][j], target <= 104`

Code Snippets

C++:

```
class Solution {
public:
bool searchMatrix(vector<vector<int>>& matrix, int target) {

}
};
```

Java:

```
class Solution {
public boolean searchMatrix(int[][] matrix, int target) {

}
}
```

Python3:

```
class Solution:
def searchMatrix(self, matrix: List[List[int]], target: int) -> bool:
```