mxn 행렬에서 값을 찾아내는 효율적인 알고리즘을 구현하라. 행렬은 왼쪽에서오른쪽, 위에서 아래 오름차순으로 정렬되어 있다.

Input: matrix = [[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],[18,21,23,26,30]], target = 5
Output: true

1.이진 검색

```
import bisect

class Solution:
    def searchMatrix(self, matrix: List[List[int]], target: int) -> bool:
        for m in matrix:
            if m[0] <= target <= m[-1]:
                i = bisect.bisect_left(m, target)
                if 0 <= i < len(m) and m[i] == target:
                    return True
        return False
```

```
import bisect

class Solution:
    def searchMatrix(self, matrix: List[List[int]], target: int) -> bool:
        if not matrix:
            return False
        row = 0
        col = len(matrix[0]) - 1
        while row <= len(matrix) - 1 and col >= 0:
            if target == matrix[row][col]:
                return True
            elif target < matrix[row][col]:
                col -= 1
            elif target > matrix[row][col]:
                row += 1
        return False
```

2.파이썬 다운

```
class Solution:
    def searchMatrix(self, matrix: List[List[int]], target: int) -> bool:
        return any([target in m for m in matrix])
```