**74. Search a 2D Matrix :-**

Medium Accepted: 1.4M Submissions: 2.8M Acceptance Rate: 48.7%

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 :-**

class Solution {

public:

    bool searchMatrix(vector<vector<int>>& matrix, int target) {

        int row=matrix.size(), col=matrix[0].size(), low=0, high=row\*col -1;

        while(low <= high){

            int mid = low + (high - low) / 2;

            if(matrix[mid/col][mid%col] == target)

                return true;

            else if(matrix[mid/col][mid%col] < target)

                low = mid + 1;

            else

                high = mid - 1;

        }

        return false;

    }

};

**T.C :- O(log(M\*N))**

**S.C :- O(1)**