

Name: Anshul

UID: 22BCS16477

Section/Group: 609(B)

Search a 2D Matrix II

Code:

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int m = matrix.size();    // Number of rows
        int n = matrix[0].size(); // Number of columns
        // Start from the top-right corner
        int row = 0;
        int col = n - 1;
        while (row < m && col >= 0) {
            if (matrix[row][col] == target) {
                return true; // Found the target
            } else if (matrix[row][col] > target) {
                col--; // Move left if current element is larger than target
            } else {
                row++; // Move down if current element is smaller than target
            }
        }
        return false; // Target not found
    }
};
```

Output:

The screenshot shows the LeetCode interface for the problem "Search a 2D Matrix II". The submission is marked as "Accepted" with 130/130 test cases passed. The user's submission details show a runtime of 47 ms (Beats 81.40%) and a memory usage of 18.59 MB (Beats 91.53%). A histogram of runtime times is displayed, with the user's solution being the fastest. The code editor shows the C++ solution code, which is the same as the one provided in the text. The test case section shows a matrix and a target value.

Accepted 130 / 130 testcases passed  
22BCS16477 Anshul submitted at Mar 18, 2025 11:36

Runtime 47 ms Beats 81.40%  
Memory 18.59 MB Beats 91.53%

47ms 81.40% 18.59MB 91.53%

47ms 81.40% 18.59MB 91.53%

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int m = matrix.size();    // Number of rows
        int n = matrix[0].size(); // Number of columns
        // Start from the top-right corner
        int row = 0;
        int col = n - 1;
        while (row < m && col >= 0) {
            if (matrix[row][col] == target) {
                return true; // Found the target
            } else if (matrix[row][col] > target) {
                col--; // Move left if current element is larger than target
            } else {
                row++; // Move down if current element is smaller than target
            }
        }
        return false; // Target not found
    }
};
```

Testcase 1

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]]