Search in 2D Matrix using Binary Search

Concept:
The problem is to search for an element in a 2D matrix where:
- Integers in each row are sorted in ascending order.
- The first integer of each row is greater than the last integer of the previous row.

Approach:
We can treat the 2D matrix as a 1D sorted array and apply binary search:
- Convert mid index into row and column using:
row = mid / m, col = mid % m
- If element at (row, col) matches target, return true.
- If smaller, move right half (low = mid + 1).
- If larger, move left half (high = mid - 1).

Time Complexity: O(log(n * m))
Space Complexity: O(1)

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& mat, int x) {
        int n = mat.size();
        int m = mat[0].size();
        int low = 0;
        int high = n * m - 1;

        while (low <= high) {
            int mid = (low + high) / 2;
            int row = mid / m, col = mid % m;

            if (mat[row][col] == x) {
                return true;
            }

            else if (mat[row][col] < x) {
                low = mid + 1;
            }

            else {
                high = mid - 1;
            }
        }

        return false;
    }
};
```