

Search In a Matrix [LeetCode](#)

Given a 2D matrix where each row and each column is sorted in ascending order, write a function to determine whether a given target element exists in the matrix.

Matrix:

13 14 23 33

39 44 51 57

58 62 67 79

89 93 98 99, target = 98

Output:

Using the Linear search approach

The target 98 is present in the matrix

Using the Binary search approach

The target 98 is present in the matrix

Approach 1: Function to search for a target using the linear search approach

- Iterate through each element in the matrix and compare it with the target.
- If the target is found, return **true**.
- If the entire matrix is traversed without finding the target, return **false**.
- **Time Complexity: $O(\text{rows} * \text{cols})$, where 'rows' is the number of rows and 'cols' is the number of columns in the matrix.**
- **Space Complexity: $O(1)$, as it uses only a constant amount of extra space for variables.**

Approach 2: Function to search for a target using the binary search approach

- Treat the 2D matrix as a sorted 1D array and apply binary search.
- Calculate the mid index and access the corresponding row and column indices in the matrix.
- If the mid element is equal to the target, return **true**.

- If the mid element is less than the target, continue the search in the right half of the matrix.
- If the mid element is greater than the target, continue the search in the left half of the matrix.
- **Time Complexity: $O(\log(\text{rows} * \text{cols}))$, where 'rows' is the number of rows and 'cols' is the number of columns in the matrix.**
- **Space Complexity: $O(1)$, as it uses only a constant amount of extra space for variables.**

Which Approach to Use?

- Use the Linear Search approach when the matrix is small or the input size is not large. It is simpler to implement and doesn't require the matrix to be sorted.
- Use the Binary Search approach when the matrix is large and the input size is significant. It takes advantage of the sorted nature of the matrix and provides better time complexity.

Better Approach

- The Binary Search approach is generally more efficient due to its logarithmic time complexity. However.