# Double Binary Search

Given a sorted matrix where each row and each column is in ascending order, your task is to locate the position of a target element within the matrix. If the element exists, return its zero-based indices (row, column). If the element does not exist, print (-1, -1).

**Input:**

- The number of rows $m$ followed by the number of columns $n$
- The sorted matrix values $(m \times n)$
- The target value

Please note that all the inputs are numbers, and it may be separated either by space or by newline. The code should be able to handle both.

**Output:**

- The indices of the result if it is found, $(-1, -1)$ otherwise.

**Sample**:

| No. | Sample Input | Sample Output |
|---|---|---|
| 1 | 3 3<br>1 2 3<br>4 5 6<br>7 8 9<br>6 | 1 2 |
| 2 | 3 3<br>1 4 7<br>9 12 15<br>100 500 1000<br>50 | -1 -1 |

**Challenge**: Find the result in $O(\log m + \log n)$ time! Use recursion!

**Additional Notes:**

- It is guaranteed that there will be no duplicates values.
- It is guaranteed that the array is in increasing order.
- It is guaranteed that the first element of a row is greater than the last element of the previous row.