

```
public int searchInsert(int[] nums, int target) {  
    if (nums == null || nums.length == 0) {  
        return -1;  
    }  
  
    int start = 0;  
    int end = nums.length - 1;  
    while (start + 1 < end) {  
        int mid = start + (end - start) / 2;  
        if (target > nums[mid]) {  
            start = mid;  
        } else {  
            end = mid;  
        }  
    }  
  
    if (nums[start] == target) {  
        return start;  
    }  
    if (nums[end] == target) {  
        return end;  
    }  
  
    if (target < nums[0]) {  
        return 0;  
    }  
    if (nums[end] < target) {  
        return end + 1;  
    }  
    return start + 1;  
}
```

搜索二维矩阵

编写一个高效的算法来搜索 $m \times n$ 矩阵中的一个目标值。

该矩阵具有以下特性：

每行中的整数从左到右排序。

每行的第一个整数大于前一行的最后一个整数。

例如，

以下矩阵：

```
[  
  [1, 3, 5, 7],  
  [10, 11, 16, 20],  
  [23, 30, 34, 50]  
]
```

给定目标值= 3，返回 true。

<https://leetcode-cn.com/problems/search-a-2d-matrix/description/>