

# Problem 1198: Find Smallest Common Element in All Rows

## Problem Information

Difficulty: Medium

Acceptance Rate: 0.00%

Paid Only: No

## Problem Description

Given an

$m \times n$

matrix

mat

where every row is sorted in

strictly

increasing

order, return

the

smallest common element

in all rows

.

If there is no common element, return

-1

.

Example 1:

Input:

```
mat = [[1,2,3,4,5],[2,4,5,8,10],[3,5,7,9,11],[1,3,5,7,9]]
```

Output:

5

Example 2:

Input:

```
mat = [[1,2,3],[2,3,4],[2,3,5]]
```

Output:

2

Constraints:

$m == mat.length$

$n == mat[i].length$

$1 \leq m, n \leq 500$

$1 \leq mat[i][j] \leq 10$

4

$mat[i]$

is sorted in strictly increasing order.

## Code Snippets

### C++:

```
class Solution {  
public:  
    int smallestCommonElement(vector<vector<int>>& mat) {  
  
    }  
};
```

### Java:

```
class Solution {  
public int smallestCommonElement(int[][] mat) {  
  
}  
}
```

### Python3:

```
class Solution:  
    def smallestCommonElement(self, mat: List[List[int]]) -> int:
```

### Python:

```
class Solution(object):  
    def smallestCommonElement(self, mat):  
        """  
        :type mat: List[List[int]]  
        :rtype: int  
        """
```

### JavaScript:

```
/**  
 * @param {number[][]} mat  
 * @return {number}  
 */
```

```
var smallestCommonElement = function(mat) {  
};
```

### TypeScript:

```
function smallestCommonElement(mat: number[][]): number {  
};
```

### C#:

```
public class Solution {  
    public int SmallestCommonElement(int[][] mat) {  
  
    }  
}
```

### C:

```
int smallestCommonElement(int** mat, int matSize, int* matColSize) {  
  
}
```

### Go:

```
func smallestCommonElement(mat [][]int) int {  
  
}
```

### Kotlin:

```
class Solution {  
    fun smallestCommonElement(mat: Array<IntArray>): Int {  
  
    }  
}
```

### Swift:

```
class Solution {  
    func smallestCommonElement(_ mat: [[Int]]) -> Int {
```

```
}
```

```
}
```

### Rust:

```
impl Solution {
    pub fn smallest_common_element(mat: Vec<Vec<i32>>) -> i32 {
        }
    }
```

### Ruby:

```
# @param {Integer[][]} mat
# @return {Integer}
def smallest_common_element(mat)

end
```

### PHP:

```
class Solution {

    /**
     * @param Integer[][] $mat
     * @return Integer
     */
    function smallestCommonElement($mat) {

    }
}
```

### Dart:

```
class Solution {
    int smallestCommonElement(List<List<int>> mat) {
        }
    }
```

### Scala:

```

object Solution {
    def smallestCommonElement(mat: Array[Array[Int]]): Int = {
        }
    }
}

```

### Elixir:

```

defmodule Solution do
  @spec smallest_common_element(mat :: [[integer]]) :: integer
  def smallest_common_element(mat) do
    end
    end

```

### Erlang:

```

-spec smallest_common_element(Mat :: [[integer()]]) -> integer().
smallest_common_element(Mat) ->
  .

```

### Racket:

```

(define/contract (smallest-common-element mat)
  (-> (listof (listof exact-integer?)) exact-integer?))

```

## Solutions

### C++ Solution:

```

/*
 * Problem: Find Smallest Common Element in All Rows
 * Difficulty: Medium
 * Tags: array, hash, sort, search
 *
 * Approach: Use two pointers or sliding window technique
 * Time Complexity: O(n) or O(n log n)
 * Space Complexity: O(n) for hash map
 */

```

```
class Solution {  
public:  
    int smallestCommonElement(vector<vector<int>>& mat) {  
  
    }  
};
```

### Java Solution:

```
/**  
 * Problem: Find Smallest Common Element in All Rows  
 * Difficulty: Medium  
 * Tags: array, hash, sort, search  
 *  
 * Approach: Use two pointers or sliding window technique  
 * Time Complexity: O(n) or O(n log n)  
 * Space Complexity: O(n) for hash map  
 */  
  
class Solution {  
public int smallestCommonElement(int[][] mat) {  
  
}  
}
```

### Python3 Solution:

```
"""  
Problem: Find Smallest Common Element in All Rows  
Difficulty: Medium  
Tags: array, hash, sort, search  
  
Approach: Use two pointers or sliding window technique  
Time Complexity: O(n) or O(n log n)  
Space Complexity: O(n) for hash map  
"""  
  
class Solution:  
    def smallestCommonElement(self, mat: List[List[int]]) -> int:  
        # TODO: Implement optimized solution  
        pass
```

### Python Solution:

```
class Solution(object):
    def smallestCommonElement(self, mat):
        """
        :type mat: List[List[int]]
        :rtype: int
        """
```

### JavaScript Solution:

```
/**
 * Problem: Find Smallest Common Element in All Rows
 * Difficulty: Medium
 * Tags: array, hash, sort, search
 *
 * Approach: Use two pointers or sliding window technique
 * Time Complexity: O(n) or O(n log n)
 * Space Complexity: O(n) for hash map
 */

/**
 * @param {number[][]} mat
 * @return {number}
 */
var smallestCommonElement = function(mat) {

};
```

### TypeScript Solution:

```
/**
 * Problem: Find Smallest Common Element in All Rows
 * Difficulty: Medium
 * Tags: array, hash, sort, search
 *
 * Approach: Use two pointers or sliding window technique
 * Time Complexity: O(n) or O(n log n)
 * Space Complexity: O(n) for hash map
 */

function smallestCommonElement(mat: number[][]): number {
```

```
};
```

### C# Solution:

```
/*
 * Problem: Find Smallest Common Element in All Rows
 * Difficulty: Medium
 * Tags: array, hash, sort, search
 *
 * Approach: Use two pointers or sliding window technique
 * Time Complexity: O(n) or O(n log n)
 * Space Complexity: O(n) for hash map
 */

public class Solution {
    public int SmallestCommonElement(int[][] mat) {

    }
}
```

### C Solution:

```
/*
 * Problem: Find Smallest Common Element in All Rows
 * Difficulty: Medium
 * Tags: array, hash, sort, search
 *
 * Approach: Use two pointers or sliding window technique
 * Time Complexity: O(n) or O(n log n)
 * Space Complexity: O(n) for hash map
 */

int smallestCommonElement(int** mat, int matSize, int* matColSize) {

}
```

### Go Solution:

```
// Problem: Find Smallest Common Element in All Rows
// Difficulty: Medium
```

```

// Tags: array, hash, sort, search
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(n) for hash map

func smallestCommonElement(mat [][]int) int {
}

```

### Kotlin Solution:

```

class Solution {
    fun smallestCommonElement(mat: Array<IntArray>): Int {
        return 0
    }
}

```

### Swift Solution:

```

class Solution {
    func smallestCommonElement(_ mat: [[Int]]) -> Int {
        return 0
    }
}

```

### Rust Solution:

```

// Problem: Find Smallest Common Element in All Rows
// Difficulty: Medium
// Tags: array, hash, sort, search
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(n) for hash map

impl Solution {
    pub fn smallest_common_element(mat: Vec<Vec<i32>>) -> i32 {
        return 0
    }
}

```

### Ruby Solution:

```
# @param {Integer[][]} mat
# @return {Integer}
def smallest_common_element(mat)

end
```

### PHP Solution:

```
class Solution {

    /**
     * @param Integer[][] $mat
     * @return Integer
     */
    function smallestCommonElement($mat) {

    }
}
```

### Dart Solution:

```
class Solution {
int smallestCommonElement(List<List<int>> mat) {

}
```

### Scala Solution:

```
object Solution {
def smallestCommonElement(mat: Array[Array[Int]]): Int = {

}
```

### Elixir Solution:

```
defmodule Solution do
@spec smallest_common_element(mat :: [[integer]]) :: integer
def smallest_common_element(mat) do
```

```
end  
end
```

### Erlang Solution:

```
-spec smallest_common_element(Mat :: [[integer()]]) -> integer().  
smallest_common_element(Mat) ->  
.
```

### Racket Solution:

```
(define/contract (smallest-common-element mat)  
(-> (listof (listof exact-integer?)) exact-integer?)  
)
```