Problem 1738: Find Kth Largest XOR Coordinate Value

Problem Information

Difficulty: Medium
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

You are given a 2D

matrix

of size

m x n

, consisting of non-negative integers. You are also given an integer

k

.

The

value

of coordinate

(a, b)

of the matrix is the XOR of all

matrix[i][j]

where

0 <= i <= a < m

and

0 <= j <= b < n

(0-indexed)

.

Find the

k

th

largest value

(1-indexed)

of all the coordinates of

matrix

.

Example 1:

Input:

matrix = [[5,2],[1,6]], k = 1

Output:

7

Explanation:

The value of coordinate (0,1) is 5 XOR 2 = 7, which is the largest value.

Example 2:

Input:

matrix = [[5,2],[1,6]], k = 2

Output:

5

Explanation:

The value of coordinate (0,0) is 5 = 5, which is the 2nd largest value.

Example 3:

Input:

matrix = [[5,2],[1,6]], k = 3

Output:

4

Explanation:

The value of coordinate (1,0) is 5 XOR 1 = 4, which is the 3rd largest value.

Constraints:

m == matrix.length

n == matrix[i].length

1 <= m, n <= 1000

0 <= matrix[i][j] <= 10

6

1 <= k <= m * n

Code Snippets

C++:

```
class Solution {
public:
int kthLargestValue(vector<vector<int>>& matrix, int k) {

}
};
```

Java:

```
class Solution {
public int kthLargestValue(int[][] matrix, int k) {

}
}
```

Python3:

```
class Solution:
def kthLargestValue(self, matrix: List[List[int]], k: int) -> int:
```

Python:

```
class Solution(object):
def kthLargestValue(self, matrix, k):
"""
:type matrix: List[List[int]]
:type k: int
:rtype: int
"""
```

JavaScript:

```
/**
* @param {number[][]} matrix
* @param {number} k
* @return {number}
*/
var kthLargestValue = function(matrix, k) {

};
```

TypeScript:

```
function kthLargestValue(matrix: number[][], k: number): number {

};
```

C#:

```
public class Solution {
public int KthLargestValue(int[][] matrix, int k) {

}
}
```

C:

```
int kthLargestValue(int** matrix, int matrixSize, int* matrixColSize, int k)
{

}
```

Go:

```
func kthLargestValue(matrix [][]int, k int) int {

}
```

Kotlin:

```
class Solution {
fun kthLargestValue(matrix: Array<IntArray>, k: Int): Int {
```

```
}
}
```

Swift:

```
class Solution {
func kthLargestValue(_ matrix: [[Int]], _ k: Int) -> Int {

}
}
```

Rust:

```
impl Solution {
pub fn kth_largest_value(matrix: Vec<Vec<i32>>, k: i32) -> i32 {

}
}
```

Ruby:

```
# @param {Integer[][]} matrix
# @param {Integer} k
# @return {Integer}
def kth_largest_value(matrix, k)

end
```

PHP:

```
class Solution {

/**
* @param Integer[][] $matrix
* @param Integer $k
* @return Integer
*/
function kthLargestValue($matrix, $k) {

}
}
```

Dart:

```
class Solution {
int kthLargestValue(List<List<int>> matrix, int k) {

}
}
```

Scala:

```
object Solution {
def kthLargestValue(matrix: Array[Array[Int]], k: Int): Int = {

}
}
```

Elixir:

```
defmodule Solution do
@spec kth_largest_value(matrix :: [[integer]], k :: integer) :: integer
def kth_largest_value(matrix, k) do

end
end
```

Erlang:

```
-spec kth_largest_value(Matrix :: [[integer()]], K :: integer()) ->
integer().
kth_largest_value(Matrix, K) ->
.
```

Racket:

```
(define/contract (kth-largest-value matrix k)
(-> (listof (listof exact-integer?)) exact-integer? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Find Kth Largest XOR Coordinate Value
* Difficulty: Medium
* Tags: array, sort, queue, heap
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public:
int kthLargestValue(vector<vector<int>>& matrix, int k) {

}
};
```

Java Solution:

```
/**
* Problem: Find Kth Largest XOR Coordinate Value
* Difficulty: Medium
* Tags: array, sort, queue, heap
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public int kthLargestValue(int[][] matrix, int k) {

}
}
```

Python3 Solution:

```
"""
Problem: Find Kth Largest XOR Coordinate Value
Difficulty: Medium
Tags: array, sort, queue, heap
```

```
Approach: Use two pointers or sliding window technique
Time Complexity: O(n) or O(n log n)
Space Complexity: O(1) to O(n) depending on approach
"""

class Solution:
def kthLargestValue(self, matrix: List[List[int]], k: int) -> int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def kthLargestValue(self, matrix, k):
"""
:type matrix: List[List[int]]
:type k: int
:rtype: int
"""
```

JavaScript Solution:

```
/**
* Problem: Find Kth Largest XOR Coordinate Value
* Difficulty: Medium
* Tags: array, sort, queue, heap
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

/**
* @param {number[][]} matrix
* @param {number} k
* @return {number}
*/
var kthLargestValue = function(matrix, k) {

};
```

TypeScript Solution:

```
/**
* Problem: Find Kth Largest XOR Coordinate Value
* Difficulty: Medium
* Tags: array, sort, queue, heap
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

function kthLargestValue(matrix: number[][], k: number): number {

};
```

C# Solution:

```
/*
* Problem: Find Kth Largest XOR Coordinate Value
* Difficulty: Medium
* Tags: array, sort, queue, heap
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

public class Solution {
public int KthLargestValue(int[][] matrix, int k) {

}
}
```

C Solution:

```
/*
* Problem: Find Kth Largest XOR Coordinate Value
* Difficulty: Medium
* Tags: array, sort, queue, heap
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
```

```
* Space Complexity: O(1) to O(n) depending on approach
*/

int kthLargestValue(int** matrix, int matrixSize, int* matrixColSize, int k)
{

}
```

Go Solution:

```
// Problem: Find Kth Largest XOR Coordinate Value
// Difficulty: Medium
// Tags: array, sort, queue, heap
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

func kthLargestValue(matrix [][]int, k int) int {

}
```

Kotlin Solution:

```
class Solution {
fun kthLargestValue(matrix: Array<IntArray>, k: Int): Int {

}
}
```

Swift Solution:

```
class Solution {
func kthLargestValue(_ matrix: [[Int]], _ k: Int) -> Int {

}
}
```

Rust Solution:

```
// Problem: Find Kth Largest XOR Coordinate Value
// Difficulty: Medium
// Tags: array, sort, queue, heap
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

impl Solution {
pub fn kth_largest_value(matrix: Vec<Vec<i32>>, k: i32) -> i32 {

}
}
```

Ruby Solution:

```
# @param {Integer[][]} matrix
# @param {Integer} k
# @return {Integer}
def kth_largest_value(matrix, k)

end
```

PHP Solution:

```
class Solution {

/**
* @param Integer[][] $matrix
* @param Integer $k
* @return Integer
*/
function kthLargestValue($matrix, $k) {

}
}
```

Dart Solution:

```
class Solution {
int kthLargestValue(List<List<int>> matrix, int k) {
```

```
}
}
```

Scala Solution:

```
object Solution {
def kthLargestValue(matrix: Array[Array[Int]], k: Int): Int = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec kth_largest_value(matrix :: [[integer]], k :: integer) :: integer
def kth_largest_value(matrix, k) do

end
end
```

Erlang Solution:

```
-spec kth_largest_value(Matrix :: [[integer()]], K :: integer()) ->
integer().
kth_largest_value(Matrix, K) ->
.
```

Racket Solution:

```
(define/contract (kth-largest-value matrix k)
(-> (listof (listof exact-integer?)) exact-integer? exact-integer?)
)
```