

3. Leftmost Column with at Least a One A row-sorted binary matrix means that all elements are 0 or 1 and each row of the matrix is sorted in non-decreasing order. Given a row-sorted binary matrix `binaryMatrix`, return the index (0-indexed) of the leftmost column with a 1 in it. If such an index does not exist, return -1. You can't access the Binary Matrix directly. You may only access the matrix using a `BinaryMatrix` interface: • `BinaryMatrix.get(row, col)` returns the element of the matrix at index (row, col) (0-indexed). • `BinaryMatrix.dimensions()` returns the dimensions of the matrix as a list of 2 elements [rows, cols], which means the matrix is rows x cols. Submissions making more than 1000 calls to `BinaryMatrix.get` will be judged Wrong Answer. Also, any solutions that attempt to circumvent the judge will result in disqualification. For custom testing purposes, the input will be the entire binary matrix `mat`. You will not have access to the binary matrix directly. Example 1: Input: `mat = [[0,0],[1,1]]` Output: 0

PROGRAM:-

```
class BinaryMatrix:
    def get(self, row: int, col: int) -> int:
        # This method will be provided by the testing environment.
        pass

    def dimensions(self) -> list:
        # This method will be provided by the testing environment.
        pass

def leftMostColumnWithOne(binaryMatrix: BinaryMatrix) -> int:
    rows, cols = binaryMatrix.dimensions()
    current_row = 0
    current_col = cols - 1
    leftmost_col = -1

    while current_row < rows and current_col >= 0:
        if binaryMatrix.get(current_row, current_col) == 1:
            leftmost_col = current_col
            current_col -= 1 # Move left
        else:
            current_row += 1 # Move down

    return leftmost_col

# Example for custom testing
class BinaryMatrixExample(BinaryMatrix):
    def __init__(self, mat):
        self.mat = mat

    def get(self, row: int, col: int) -> int:
        return self.mat[row][col]

    def dimensions(self) -> list:
        return [len(self.mat), len(self.mat[0])]

# Example usage
mat = [[0, 0], [1, 1]]
```

```
binaryMatrix = BinaryMatrixExample(mat)
print(leftMostColumnWithOne(binaryMatrix)) # Output: 0
```

OUTPUT:-



```
0
=== Code Execution Successful ===
```

TIME COMPLEXITY:- $O(\text{num})$