Previous (/articles/reorganized-string/) Next (/articles/max-chunks-to-make-sorted-ii/)

766. Toeplitz Matrix (/problems/toeplitz-matrix/)

tings/107/384/?return=/articles/toeplitz-matrix/) (/ratings/107/384/?return=/articles/toeplitz-matrix/) (/ratings/107/384/?return=/articles/toeplitz-matrix/)

Average Rating: 4 (6 votes)

Jan. 20, 2018 | 5K views

A matrix is *Toeplitz* if every diagonal from top-left to bottom-right has the same element.

Now given an `M x N` matrix, return `True` if and only if the matrix is *Toeplitz*.

Example 1:

```
Input: matrix = [[1,2,3,4],[5,1,2,3],[9,5,1,2]]
Output: True
Explanation:
1234
5123
9512

In the above grid, the diagonals are "[9]", "[5, 5]", "[1, 1, 1]", "[2, 2, 2]", "[3, 3]", "[4]",
```

Example 2:

```
Input: matrix = [[1,2],[2,2]]
Output: False
Explanation:
The diagonal "[1, 2]" has different elements.
```

Note:

1. `matrix` will be a 2D array of integers.
2. `matrix` will have a number of rows and columns in range `[1, 20]`.
3. `matrix[i][j]` will be integers in range `[0, 99]`.

Approach #1: Group by Category [Accepted]

Intuition and Algorithm

We ask what feature makes two coordinates `(r1, c1)` and `(r2, c2)` belong to the same diagonal?

It turns out two coordinates are on the same diagonal if and only if `r1 - c1 == r2 - c2`.

This leads to the following idea: remember the value of that diagonal as `groups[r-c]`. If we see a mismatch, the matrix is not Toeplitz; otherwise it is.

Java Python Copy

```
class Solution {
    public boolean isToeplitzMatrix(int[][] matrix) {
        Map<Integer, Integer> groups = new HashMap();
        for (int r = 0; r < matrix.length; ++r) {
            for (int c = 0; c < matrix[0].length; ++c) {
                if (!groups.containsKey(r-c))
                    groups.put(r-c, matrix[r][c]);
                else if (groups.get(r-c) != matrix[r][c])
                    return False;
            }
        }
        return True;
    }
}
```

Complexity Analysis

- Time Complexity: $O(M * N)$. (Recall in the problem statement that M, N are the number of rows and columns in `matrix`.)
- Space Complexity: $O(M * N)$.

Approach #2: Compare With Top-Left Neighbor [Accepted]

Intuition and Algorithm

For each diagonal with elements in order $a_1, a_2, a_3, \dots, a_k$, we can check $a_1 = a_2, a_2 = a_3, \dots, a_{k-1} = a_k$. The matrix is *Toeplitz* if and only if all of these conditions are true for all (top-left to bottom-right) diagonals.

Every element belongs to some diagonal, and it's previous element (if it exists) is it's top-left neighbor. Thus, for the square `(r, c)`, we only need to check `r == 0 OR c == 0 OR matrix[r-1][c-1] == matrix[r][c]`.

Java Python Copy

```
class Solution {
    public boolean isToeplitzMatrix(int[][] matrix) {
        for (int r = 0; r < matrix.length; ++r)
            for (int c = 0; c < matrix[0].length; ++c)
                if (r > 0 && c > 0 && matrix[r-1][c-1] != matrix[r][c])
                    return false;
        return true;
    }
}
```

Complexity Analysis

- Time Complexity: $O(M * N)$, as defined in the problem statement.
- Space Complexity: $O(1)$.

Analysis written by: @awice (https://leetcode.com/awice).

Rate this article:

(/ratings/107/384/?return=/articles/toeplitz-matrix/) (/ratings/107/384/?return=/articles/toeplitz-m

Previous (/articles/reorganized-string/)

Next (/articles/max-chunks-to-make-sorted-ii/)

Join the conversation

Login to Reply

C

chairco commented 2 weeks ago

(https://discuss.leetcode.com/user/chairco)

python

```
class Solution:
    def isToeplitzMatrix(self, matrix):
        """
        :type matrix: List[List[int]]
        :rtype: bool
        """
        row = len(matrix[0])
        col = len(matrix)

        candidate = matrix[0]
        for i in range(1, col):
            move = 1
            for j in range(move, row):
                if candidate[j-1] != matrix[i][j]:
                    return False
            candidate = matrix[i]
            move += 1
        return True
```

M

Metaphysics commented 3 weeks ago

(https://discuss.leetcode.com/user/metaphysics)

C#
public class Solution {
public bool IsToeplitzMatrix(int[,] matrix) {
int col=matrix.GetLength(1);
for(int i=0;i<matrix.Length-col;i++)
{
if((i+1)%col!=0 && matrix[i/col,i%col]!=matrix[(i+col+1)/col,(i+col+1)%col])
{
return false;
}
}
return true;
}
}

B

boloutare commented last month

(https://discuss.leetcode.com/user/boloutare)

```
class Solution:
    def isToeplitzMatrix(self, matrix):
        """
        :type matrix: List[List[int]]
        :rtype: bool
        """
        for j in range(1, len(matrix)):
            if matrix[j-1][:len(matrix[j])-1] != matrix[j][1:]:
                return False
        return True
```

B

boloutare commented last month

(https://discuss.leetcode.com/user/boloutare)

`class Solution: def isToeplitzMatrix(self, matrix): """ :type matrix: List[List[int]] :rtype: bool """ for j in range(1, len(matrix)): if matrix[j-1][:len(matrix[j])-1] != matrix[j][1:]: return False return True`

P

(https://discuss.leetcode.com/user/prakashmanwani)

prakashmanwani commented last month

class Solution {
public boolean isToeplitzMatrix(int[][] matrix) {
System.out.println(matrix.length);

```
    int rows = matrix.length;
    int columns = matrix[0].length;
    for (int i=1;i<rows;i++){
        for (int j =1;j<columns;j++){
        if (matrix[i-1][j-1]!=matrix[i][j])return false;
        }
    }
    return true;

}
```

}

S

(https://discuss.leetcode.com/user/shermm)

SherMM commented last month

```
def isToeplitzMatrix(self, matrix):
    """
    :type matrix: List[List[int]]
    :rtype: bool
    """
    r, c = len(matrix), len(matrix[0])
    for i in range(r-1):
        for j in range(c-1):
            if matrix[i+1][j+1] != matrix[i][j]:
                return False
    return True
```

You only need to iterate to next to last row and column, since single-value items will be Toeplitz.

(https://discuss.leetcode.com/user/nevillealee)

Nevillealee commented 2 months ago

Javascript
/**

- @param {number[][]} matrix
- @return {boolean}
  */
  var isToeplitzMatrix = function(matrix) {
  for(var r = 1; r < matrix.length; r++) {
  for(var c = 1; c < matrix[0].length; c++) {
  if(matrix[r][c] != matrix[r-1][c-1]) {
  return false;
  }
  }
  }
  return true;
  };

M

(https://discuss.leetcode.com/user/marirod)

marirod commented 2 months ago

@StefanPochmann (https://discuss.leetcode.com/uid/591) Indeed, thanks for the fast reply!

(https://discuss.leetcode.com/user/awice)

awice commented 2 months ago

@StefanPochmann (https://discuss.leetcode.com/uid/591) Thanks, corrected. Funny because I write the problem statements ;(Though with that definition of N it was correct.

(https://discuss.leetcode.com/user/stefanpochmann)

StefanPochmann commented 2 months ago

@marirod (https://discuss.leetcode.com/uid/389430) Should say neither O(N) nor O(N^2) but O(MN), I guess @awice (https://discuss.leetcode.com/uid/71269) overlooked that M and N are already defined in the problem.

View original thread (https://discuss.leetcode.com/topic/117862)

Load more comments...

Contact Us (/support/) | Frequently Asked Questions (/faq/) | Terms of Service (/terms/) | Privacy Policy (/privacy/)

United States (/region/)