

CodeCheck Report: trainingUXHEMH-6ZB

Test Name:

[Check out Codility training tasks](#)

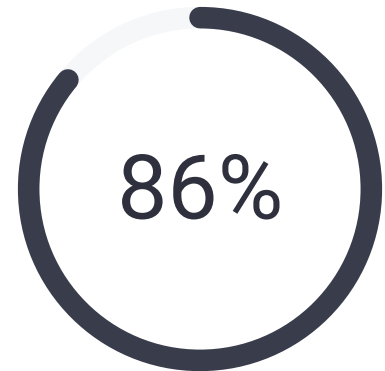
Summary

Timeline

Tasks summary

Task	Time spent	Score
CountriesCount Python	16 min	86%

Total score



Tasks Details

Hard	1. CountriesCount	Task Score	Correctness	Performance
	Count the number of different countries that a map contains.	86%	100%	66%

Task description

A rectangular map consisting of N rows and M columns of square areas is given. Each area is painted with a certain color.

Two areas on the map *belong to the same country* if the following conditions are met:

- they have the same color;
- it is possible to travel from one area to the other orthogonally (that is, by moving only north, south, west or east) without moving over areas of a different color.

The map can be described by a zero-indexed matrix A consisting of N rows and M columns of integers. The color of each area is described by the corresponding element of the matrix. Two areas have the same color if and only if

Solution

Programming language used: Python

Total time used: 16 minutes ?

Effective time used: 16 minutes ?

Notes: *not defined yet*

Task timeline

?

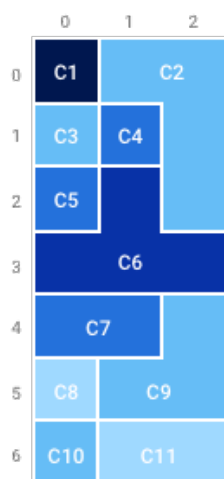


their corresponding matrix elements have the same value.

For example, consider the following matrix A consisting of seven rows and three columns:

```
A[0][0] = 5   A[0][1] = 4   A[0][2] = 4
A[1][0] = 4   A[1][1] = 3   A[1][2] = 4
A[2][0] = 3   A[2][1] = 2   A[2][2] = 4
A[3][0] = 2   A[3][1] = 2   A[3][2] = 2
A[4][0] = 3   A[4][1] = 3   A[4][2] = 4
A[5][0] = 1   A[5][1] = 4   A[5][2] = 4
A[6][0] = 4   A[6][1] = 1   A[6][2] = 1
```

Matrix A describes a map that is colored with five colors. The areas on the map belong to eleven different countries (C1–C11), as shown in the following figure:



Write a function

```
def solution(A)
```

that, given a zero-indexed matrix A consisting of N rows and M columns of integers, returns the number of different countries to which the areas of the map described by matrix A belong.

For example, given matrix A consisting of seven rows and three columns corresponding to the example above, the function should return 11.

Write an **efficient** algorithm for the following assumptions:

- N and M are integers within the range [1..300,000];
- the number of elements in matrix A is within the range [1..300,000];
- each element of matrix A is an integer within the range [−1,000,000,000..1,000,000,000].

Copyright 2009–2023 by Codility Limited. All Rights Reserved.

Unauthorized copying, publication or disclosure prohibited.

07:10:44

07:26:14

Code: 07:26:13 UTC, py, [show code in pop-up](#)
final, score: 86

```
1 # you can write to stdout for debugging p
2 # print("this is a debug message")
3
4 def solution(A):
5     # Implement your solution here
6     n = len(A)
7     if n == 0:
8         return 0
9     m = len(A[0])
10    visited = [[False]*m for _ in range(n)
11    countries = 0
12
13    def dfs(i, j, color):
14        if i<0 or i>=n or j<0 or j>=m or
15            return
16        visited[i][j] = True
17        dfs(i+1, j, color)
18        dfs(i-1, j, color)
19        dfs(i, j+1, color)
20        dfs(i, j-1, color)
21
22    for i in range(n):
23        for j in range(m):
24            if not visited[i][j]:
25                countries += 1
26                dfs(i, j, A[i][j])
27
28    return countries
29
```

Analysis summary

The following issues have been detected: runtime errors.

Analysis

Detected time complexity: **$O(N \cdot M)$**

expand all	Example tests
▶ example	✓ OK
expand all	Correctness tests
▶ small_1x1	✓ OK
▶ positive_negative_zeros	✓ OK
▶ small_2x2	✓ OK
▶ small_3x3	✓ OK
▶ matrix_12x10	✓ OK
▶ matrix_10x10_labyrinth	✓ OK
▶ matrix_wide	✓ OK
▶ large_numbers	✓ OK

▶	anti_heuristics	✓ OK
expand all Performance tests		
▶	medium_matrix	✓ OK
▶	medium_square_matrix	✓ OK
▶	medium_horizontal_matrix	✓ OK
▶	large_square_one_country	✗ RUNTIME ERROR tested program terminated with exit code 1
▶	max_matrix	✓ OK
▼	max_one_country	✗ RUNTIME ERROR tested program terminated with exit code 1
<hr/>		
1.	0.112 RUNTIME ERROR , tested program terminated with s exit code 1 stderr: Traceback (most recent call last): File "exec.py", line 118, in <module> main() File "exec.py", line 94, in main result = solution(A, B)	