All Topics (/dashboard/) / Topic - Graphs (/view_topic/7/) / Test - homework - Graphs (/view_test/140/)
/ Problem - Zombie Clusters (/view_test_problem/140/130/) (✔ **Completed**)

Problem Statement (/view_test_problem/140/130/) Previous Submissions (/view_all_submissions_for_test_problem/140/130/)

🔒 Editorial Top Submissions (/view_top_submissions/140/130/30/)

− Problem Description

There are zombies in Seattle. Liv and Ravi are trying to track them down to find out who is creating new zombies in an effort to prevent an apocalypse. Other than the patient-zero zombies (who became so by mixing MaxRager and tainted Utopium), new people only become zombies after being scratched by an existing zombie. Zombiism is transitive. This means that if zombie *0* knows zombie *1* and zombie *1* knows zombie *2*, then zombie *0* is connected to zombie *2* by way of knowing zombie *1*. A zombie *cluster* is a group of zombies who are directly or indirectly linked through the other zombies they know, such as the one who scratched them or supplies who them with brains.

The diagram showing connectedness will be made up of a number of binary strings, characters 0 or 1. Each of the characters in the string represents whether the zombie associated with a row element is connected to the zombie at that character's index. For instance, a zombie 0 with a connectedness string '110' is connected to zombies 0 (itself) and zombie 1, but not to zombie 2. The complete matrix of zombie connectedness is:

```
        110
        110
        001
```

Zombies 0 and 1 are connected. Zombie 2 is not.

Your task is to determine the number of connected groups of zombies, or clusters, in a given matrix.

Note: Method signatures may vary depending on the requirements of your chosen language.

Function Description
Complete the function *zombieCluster* in the editor below. The function must return an integer representing the number of zombie clusters counted.

zombieCluster has the following parameter(s):
zombies[z0,...zn-1]: an array of strings of binary digits *zi* representing connectedness of zombies

Constraints

- *1 ≤ n ≤ 300*
- *0 ≤ i < n*
- *|zombies| = n*
- Each *zi* contains a binary string of *n* zeroes and ones. It is a square matrix.

Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer *n*, the size of the square zombie association matrix, *zombies*.
The next *n* lines each contain a binary string of length *n* describing a row in the matrix, *zi* where 0 ≤ i < n.

Sample Case 0

Sample Input 0

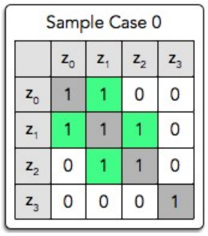
```
4
1100
1110
0110
0001
```

Sample Output 0

```
2
```

In the diagram below, the squares highlighting a known connection between two different zombies are highlighted in green. Because each zombie is already aware that they are personally a zombie, those are highlighted in grey.

Explanation 0



Sample Case 0

	z_0	z_1	z_2	z_3
z_0	1	1	0	0
z_1	1	1	1	0
z_2	0	1	1	0
z_3	0	0	0	1

We have n = 4 zombies numbered z0 through z3. There are 2 pairs of zombies who directly know each another: (z0, z1) and (z1, z2). Because of zombiism's transitive property, the set of zombies {z0, z1, z2} is considered to be a single zombie cluster. The remaining zombie, z3, doesn't know any other zombies and is considered to be his own, separate zombie cluster ({z3}). This gives us a total of 2 zombie clusters.

Sample Case 1

Sample Input 1

```
5
10000
01000
00100
00010
00001
```

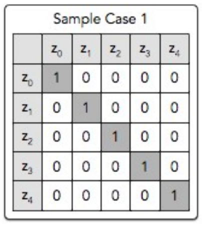
Sample Output 1

```
5
```

Explanation

In the diagram below, the squares highlighting a known connection between two different zombies are highlighted in green. Because each zombie is already aware that they are personally a zombie, those are highlighted in grey.

Explanation 1:



Sample Case 1

	z_0	z_1	z_2	z_3	z_4
z_0	1	0	0	0	0
z_1	0	1	0	0	0
z_2	0	0	1	0	0
z_3	0	0	0	1	0
z_4	0	0	0	0	1

No zombie knows who any other zombie is, so they each form their own zombie cluster: {z0}, {z1}, {z2}, {z3}, and {z4}. This means we have 5 zombie clusters, so we print 5 on a new line.

− Code Editor

Python 3 (python 3.6.6) Theme: Light Reset My Code Auto Complete On Auto Complete Off

□ Show Input/Output Code

```
# Complete the zombieCluster function below.
def explore(i, visited, zombies):
    if i in visited:
        return

    visited.add(i)
    for j in range(len(zombies)):
        if zombies[i][j] == '1':
            explore(j, visited, zombies)

def zombieCluster(zombies):
    visited = set()
    no_of_zombie_group = 0
    for i in range(len(zombies)):
        if i not in visited:
            explore(i, visited, zombies)
            no_of_zombie_group += 1
    return no_of_zombie_group
```

□ **Run against custom Input**

Quick Test (Takes ~15s)

Full Test (Takes ~45s)

- Quick Test Result

No result to show!

- Full Test Result

No result to show!