# SEARCH

[문제 설명]

1과 0으로 이루어진 2차원 배열이 주어지면, 1으로 연결되어 있는 부분을 찾아야 합니다.

상하좌우가 모두 0으로 이루어져 있다면, 분리되어 있는 곳으로 판단할 수 있습니다.

1으로 이루어진 구역의 개수를 측정해서 반환해주세요.

예를 들어,

grid = [

["1","1","0","0","0"],

["1","1","0","0","0"],

["0","0","1","0","0"],

["0","0","0","1","1"]

]

와 같은 배열이 주어진다면, 1로 구분되는 3개의 구역이 존재하는 것입니다.

고로 3을 결과값으로 반환해주시면 됩니다.

[제한 사항]

- array의 크기는 1 \* 1 이상 300 \* 300 이하입니다.

- array 안에는 항상 0 또는 1이 포함되어 있습니다.

[입력 형식]

- 2차원 배열 grid가 주어집니다.

[출력 형식]

- 1으로 구분되는 구역의 개수를 세서 반환해주세요.

제한 시간: 20분

문제 유형: BFS / DFS

난이도: 중

매개변수

|  |
| --- |
| grid |
| Array / list |

리턴타입

|  |
| --- |
| 리턴타입 |
| int |

초기코드 Python

def solution(grid):

  '''

:param grid: list

  :return: int

  '''

  answer = 0

  return answer

초기코드JavaScript

/\*\*

 \* @param grid {array}

 \* @return int

 \*/

function solution(grid) {

    return 0

}

테스트 케이스

예제용

|  |
| --- |
| 입력값 grid |
| [ ["1","1","1","1","0"],  ["1","1","0","1","0"],  ["1","1","0","0","0"],  ["0","0","0","0","0"]] |

|  |
| --- |
| 출력값 |
| 1 |

채점용

|  |  |
| --- | --- |
| 입력값 grid | 출력값 |
| [["1","1","0","1","0"],  ["1","1","0","1","0"],  ["1","1","0","0","0"],  ["0","0","0","0","0"]] | **2** |

|  |  |
| --- | --- |
| 입력값 grid | 출력값 |
| [["1","1","0","1","0"],  ["1","1","0","1","0"],  ["1","1","0","0","0"],  ["0","0","0","1","0"]] | **3** |

|  |  |
| --- | --- |
| 입력값 grid | 출력값 |
| [["1","1","0","1","0"],  ["1","1","0","1","0"],  ["0","0","0","0","0"],  ["0","1","0","1","0"]] | **4** |

|  |  |
| --- | --- |
| 입력값 grid | 출력값 |
| [["1","1","1","1","1"],  ["1","1","1","1","1"],  ["1","1","1","1","1"],  ["1","1","1","1","1"]] | **1** |

|  |  |
| --- | --- |
| 입력값 prices | 출력값 |
| [["0","0","0","0","0"],  ["0","0","0","0","0"],  ["0","0","0","0","0"],  ["0","0","0","0","0"]] | **0** |