

퍼즐 조각 채우기

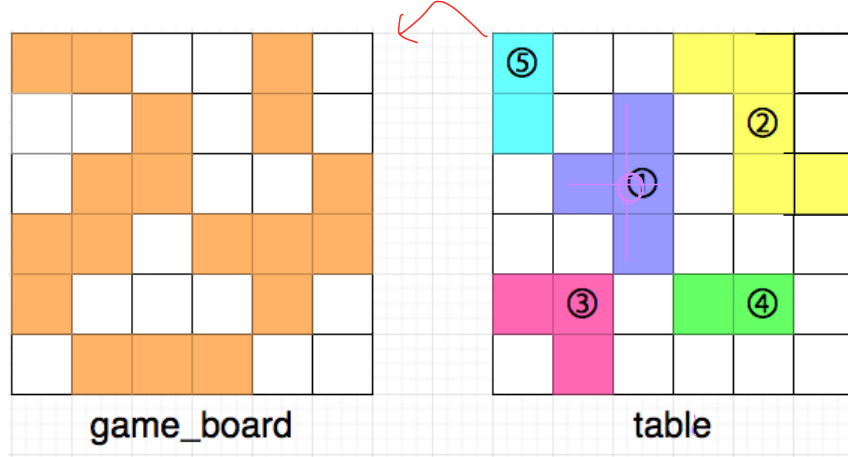
Python3

문제 설명

테이블 위에 놓인 퍼즐 조각을 게임 보드의 빈 공간에 적절히 올려놓으려 합니다. 게임 보드와 테이블은 모두 각 칸이 1×1 크기인 정사각 격자 모양입니다. 이때, 다음 규칙에 따라 테이블 위에 놓인 퍼즐 조각을 게임 보드의 빈칸에 채우면 됩니다.

- 조각은 한 번에 하나씩 채워 넣습니다.
- 조각을 회전시킬 수 있습니다.
- 조각을 뒤집을 수는 없습니다.
- 게임 보드에 새로 채워 넣은 퍼즐 조각과 인접한 칸이 비어있으면 안 됩니다.

다음은 퍼즐 조각을 채우는 예시입니다.



난이도가 상당히 높음.

① 각각 도형을 구해서 구현함

② 도형 회전을 하며 잘맞게 있으면 채택함.

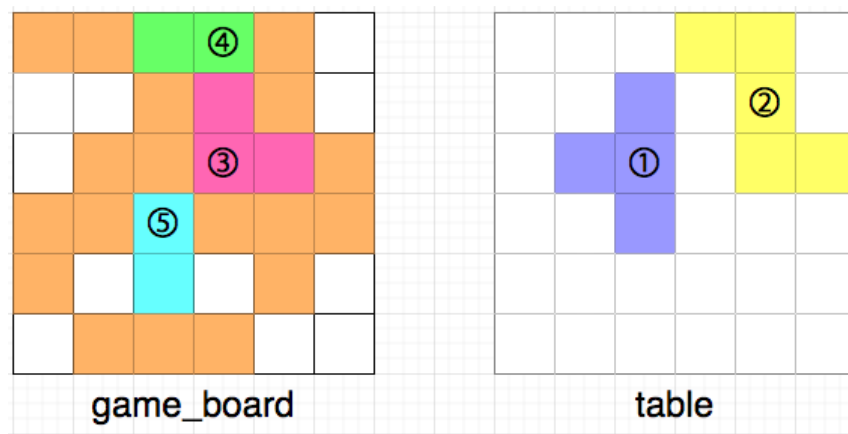


비교 (x) 방법을 찾아야 함

함수가 다 '0' 이면 해석에 0을 주지않고
되자않고.

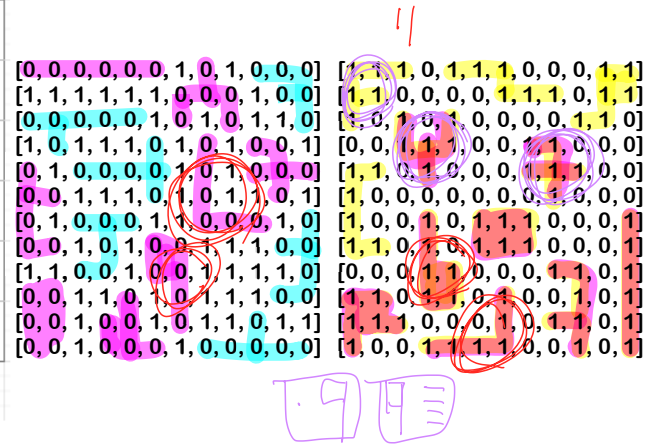
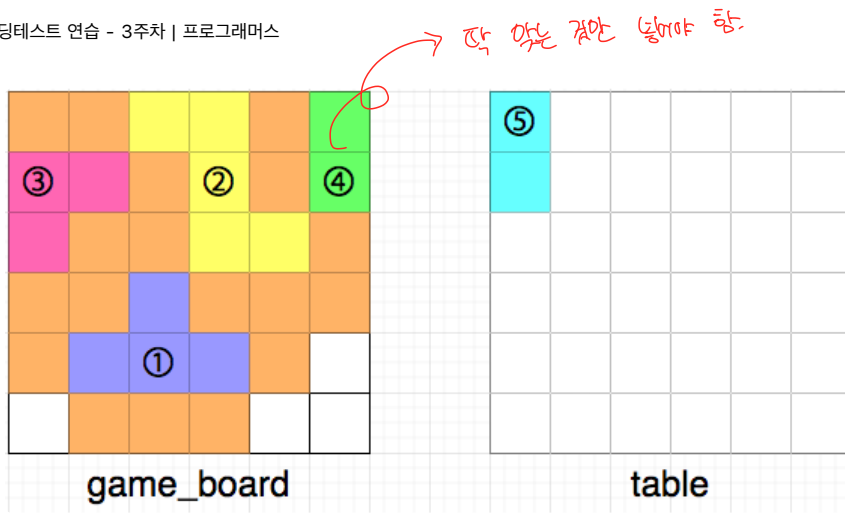
위 그림에서 왼쪽은 현재 게임 보드의 상태를, 오른쪽은 테이블 위에 놓인 퍼즐 조각들을 나타냅니다. 테이블 위에 놓인 퍼즐 조각들 또한 마찬가지로 [상,하,좌,우]로 인접해 붙어있는 경우는 없으며, 흰 칸은 퍼즐이 놓이지 않은 빈 공간을 나타냅니다. 모든 퍼즐 조각은 격자 칸에 딱 맞게 놓여있으며, 격자 칸을 벗어나거나, 걸쳐 있는 등 잘못 놓인 경우는 없습니다.

이때, 아래 그림과 같이 3,4,5번 조각을 격자 칸에 놓으면 규칙에 어긋나므로 불가능한 경우입니다.



- 3번 조각을 놓고 4번 조각을 놓기 전에 위쪽으로 인접한 칸에 빈칸이 생깁니다.
- 5번 조각의 양 옆으로 인접한 칸에 빈칸이 생깁니다.

다음은 규칙에 맞게 최대한 많은 조각을 게임 보드에 채워 넣은 모습입니다.



최대한 많은 조각을 채워 넣으면 총 14칸을 채울 수 있습니다.

현재 게임 보드의 상태 `game_board`, 테이블 위에 놓인 퍼즐 조각의 상태 `table`이 매개변수로 주어집니다. 규칙에 맞게 최대한 많은 퍼즐 조각을 채워 넣을 경우, 총 몇 칸을 채울 수 있는지 return 하도록 solution 함수를 완성해주세요.

제한사항

- `3 ≤ game_board`의 행 길이 ≤ 50
- `game_board`의 각 열 길이 = `game_board`의 행 길이
 - 즉, 게임 보드는 정사각 격자 모양입니다.
 - `game_board`의 모든 원소는 0 또는 1입니다.
 - 0은 빈칸, 1은 이미 채워진 칸을 나타냅니다.
 - 퍼즐 조각이 놓일 빈칸은 1×1 크기 정사각형이 최소 1개에서 최대 6개까지 연결된 형태로만 주어집니다.
- `table`의 행 길이 = `game_board`의 행 길이
- `table`의 각 열 길이 = `table`의 행 길이
 - 즉, 테이블은 `game_board`와 같은 크기의 정사각 격자 모양입니다.
 - `table`의 모든 원소는 0 또는 1입니다.
 - 0은 빈칸, 1은 조각이 놓인 칸을 나타냅니다.
 - 퍼즐 조각은 1×1 크기 정사각형이 최소 1개에서 최대 6개까지 연결된 형태로만 주어집니다.
- `game_board`에는 반드시 하나 이상의 빈칸이 있습니다.
- `table`에는 반드시 하나 이상의 블록이 놓여 있습니다.

입출력 예

game_board	table	result
[[1,0,0,1,0],[0,0,1,0,1,0],[0,1,1,0,0,1],[1,1,0,1,1,1],[1,0,0,0,1,0],[0,1,1,1,0,0]]	[[1,0,0,1,1,0],[1,0,1,0,1,0],[0,1,1,0,1,1],[0,0,1,0,0,0],[1,1,0,1,1,0],[0,1,0,0,0,0]]	14
[[0,0,0],[1,1,0],[1,1,1]]	[[1,1,1],[1,0,0],[0,0,0]]	0

입출력 예 설명

입출력 예 #1

입력은 다음과 같은 형태이며, 문제의 예시와 같습니다.

1	1	0	0	1	0				1	0	0	1	1	0
0	0	1	0	1	0				1	0	1	0	1	0
0	1	1	0	0	1				0	1	1	0	1	1
1	1	0	1	1	1				0	0	1	0	0	0
1	0	0	0	1	0				1	1	0	1	1	0
0	1	1	1	0	0				0	1	0	0	0	0
game_board									table					

입출력 예 #2

블록의 회전은 가능하지만, 뒤집을 수는 없습니다.

solution.py

```
1 def solution(game_board, table):
2     answer = -1
3     return answer
```

실행 결과

실행 결과가 여기에 표시됩니다.

초기화

코드 실행

제출 후 채점하기