Gold Mine - 2

1. You are given a number n, representing the number of rows.
2. You are given a number m, representing the number of columns.
3. You are given n*m numbers, representing elements of 2d array a, which represents a gold mine.
4. You are allowed to take one step left, right, up or down from your current position.
5. You can't visit a cell with 0 gold and the same cell more than once.
6. Each cell has a value that is the amount of gold available in the cell.
7. You are required to identify the maximum amount of gold that can be dug out from the mine if you start and stop collecting gold from any position in the grid that has some gold.

Note -> Check out the question video and write the recursive code as it is intended without changing signature. The judge can't force you but intends you to teach a concept.

A number n
A number m
e11
e12..
e12
e22..
m*n numbers

1 <= n <= 10
1 <= m <= 10
0 <= e1, e2, .. n * m elements <= 1000

```
matrix = [[0, 1, 4, 2, 8, 2],
          [4, 3, 6, 5, 0, 4],
          [1, 2, 4, 1, 4, 6],
          [2, 0, 7, 3, 2, 2],
          [3, 1, 5, 9, 2, 4],
          [2, 7, 0, 8, 5, 1]]
```

```
def helper(matrix, i, j, visited):
    if i < 0 or j < 0 or i >= len(matrix) or j >= len(matrix[0]) or
visited[i][j] == True or matrix[i][j] == 0:
        return 0
    visited[i][j] = True
    up = helper(matrix, i - 1, j, visited)
    down = helper(matrix, i + 1, j, visited)
    left = helper(matrix, i, j - 1, visited)
    right = helper(matrix, i, j + 1, visited)
    return matrix[i][j] + up + down + left + right
```

```
def findGold(matrix):
    visited = [[False] * len(matrix[0]) for i in range(len(matrix))]
    ans = 0
    for i in range(len(matrix)):
        for j in range(len(matrix[0])):
            if matrix[i][j] != 0:
                temp = helper(matrix, i, j, visited)
                ans = max(ans, temp)
    return ans
```