PYTHON ASSIGNMENT

MAKE A MOVE TO PYTHON

ASSIGNMENTS

SUBMITTED TO

YASHIKA KHATRI

By: ANKUR SINGH

TASK: RIVER GAME

As a Developer write a program for matcing following constraints:

You are given a two-dimensional array (matrix) of potentially unequal height and width

containing only 0s and 1s. Each 0 represents land, and each 1 represents part of a river.

A river consists of any number of 1s that are either horizontally or vertically adjacent

(but not diagonally adjacent). The number of adjacent 1s forming a river determine its

size. Write a function that returns an array of the sizes of all rivers represented in the

input matrix. Note that these sizes do not need to be in any particular order.

```
class River(object):
    def __init__(self, numOfElements):
        self.n = numOfElements
        self.parents = [0 for _ in range(numOfElements)]
        self.rank = [0 for _ in range(numOfElements)]
        self.makeSet()

    def makeSet(self):
        for i in range(self.n):
            self.parents[i] = i

    def union(self, x, y):
        parentX = self.find(x)
        parentY = self.find(y)
        if parentX == parentY:
            return
        if self.rank[parentX] > self.rank[parentY]:
            self.parents[parentY] = parentX
        elif self.rank[parentX] < self.rank[parentY]:
            self.parents[parentX] = parentY
        else:
            self.parents[parentX] = parentY
            self.rank[parentY] += 1

    def find(self, x):
        parentX = self.parents[x]
        if x != parentX:
            parentX = self.find(parentX)
        return parentX


from collections import defaultdict
def riverSizes(matrix):
    if not matrix:
        return []
    rowCount, colCount = len(matrix), len(matrix[0])
    djs = River(rowCount * colCount)
    for i in range(rowCount):
        for j in range(colCount):
            val = matrix[i][j]
            if val == 0:
                continue

    if i + 1 < rowCount and matrix[i + 1][j] == 1:
        djs.union(i * (colCount) + j, (i + 1) * (colCount) + j)

    if i - 1 >= 0 and matrix[i - 1][j] == 1:
        djs.union(i * (colCount) + j, (i - 1) * (colCount) + j)

    if j + 1 < colCount and matrix[i][j + 1] == 1:
        djs.union(i * (colCount) + j, (i) * (colCount) + j + 1)

    if j - 1 >= 0 and matrix[i][j - 1] == 1:
```

```
    djs.union(i * (colCount) + j, (i) * (colCount) + j - 1)

ilands = defaultdict(int)
for i in range(rowCount):
    for j in range(colCount):
        if matrix[i][j] == 1:
            val = i * colCount + j
            parent = djs.find(val)
            ilands[parent] += 1

return ilands.values()
```