860 · Number of Distinct Islands

Description

Given a non-empty 2D array `grid` of 0's and 1's, an island is a group of `1`'s (representing land) connected 4-directionally (horizontal or vertical). You may assume all four edges of the grid are surrounded by water.

Count the number of distinct islands. An island is considered to be the same as another if and only if one island has the same shape as another island (and not rotated or reflected).

Notice that:

```
11
1
```

and

```
 1
11
```

are considered different island, because we do not consider reflection / rotation.

The length of each dimension in the given `grid` does not exceed `50`.

Example

Example 1:

```
Input:
  [
    [1,1,0,0,1],
    [1,0,0,0,0],
    [1,1,0,0,1],
    [0,1,0,1,1]
  ]
Output: 3
Explanation:
  11   1    1
  1        11
  11
   1
```

Example 2:

```
Input:
  [
    [1,1,0,0,0],
    [1,1,0,0,0],
    [0,0,0,1,1],
    [0,0,0,1,1]
  ]
Output: 1
```

Tags

Company

Amazon

```
class Solution:
    """
    @param grid: a list of lists of integers
    @return: return an integer, denote the number of distinct islands
    """
    def numberofDistinctIslands(self, grid):
        # write your code here
        hashSet = set()
        for i in range(len(grid)):
            for j in range(len(grid[0])):
                if grid[i][j]==1:
                    path = self.dfs(grid,i,j,'x')
                    hashSet.add(path)
        return len(hashSet)



    def dfs(self,grid,r,c,psf):
        if r<0 or c<0 or r>=len(grid) or c>=len(grid[0]) or grid[r][c]==0:
            return '#'
        grid[r][c]=0
        up = self.dfs(grid,r-1,c,'u')
        right = self.dfs(grid,r,c+1,'r')
        down = self.dfs(grid,r+1,c,'d')
        left = self.dfs(grid,r,c-1,'l')
        return psf+up+right+down+left
```