

695. Max Area of Island

Medium

You are given an $m \times n$ binary matrix grid. 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.

The area of an island is the number of cells with a value 1 in the island.

Return the maximum area of an island in grid. If there is no island, return 0.

```
class Solution:
    def maxAreaOfIsland(self, grid: List[List[int]]) -> int:
        if not grid:
            return 0
        rows, cols = len(grid), len(grid[0])
        visited = set()
        maxArea = 0

        def dfs(r, c):
            if ((r,c) in visited or
                r not in range(rows) or
                c not in range(cols) or
                grid[r][c] == 0 ):
                return 0
            visited.add((r,c))
            return 1 + dfs(r+1,c) + dfs(r-1,c) + dfs(r,c+1) + dfs(r,c-1)

        for row in range(rows):
            for col in range(cols):
                if (grid[row][col] == 1 and
                    (row, col) not in visited):
                    island_area = dfs(row, col)
                    maxArea = max(maxArea, island_area)

        return maxArea
```

```
class Solution:
    def maxAreaOfIsland(self, grid: List[List[int]]) -> int:
        rows, cols = len(grid), len(grid[0])
        self.max_area = 0
        visited = set()

        def bfs(r, c):
            area = 1
            q = collections.deque()
            q.append((r,c))
            directions = [(1,0), (-1,0), (0,1), (0,-1)]
            visited.add((r, c))
            while q:
                qr, qc = q.popleft()
                for dr, dc in directions:
                    row, col = qr + dr, qc + dc
                    if (row in range(rows) and
                        col in range(cols) and
                        (row, col) not in visited and
                        grid[row][col] == 1):
                        visited.add((row, col))
                        q.append((row, col))
                        area += 1
            self.max_area = max(self.max_area, area)

        for row in range(rows):
            for col in range(cols):
                if grid[row][col] == 1 and (row, col) is not visited:
                    bfs(row, col)

        return self.max_area
```

This BFS solution is much slower than the DFS solution