```
class Solution {
    private int area;
    public int maxAreaOfIsland(int[][] grid) {
        int max= 0;
        for(int i=0; i<grid.length; i++){
            for(int j=0; j<grid[0].length;j++){
                if(grid[i][j]==1){
                    area=0;
                    dfs(grid, i, j);
                    if(area>max){
                        max=area;
                    }
                }
            }
        }
        return max;
    }

    private void dfs(int [][]grid, int i, int j){
        if(grid[i][j]==1){
            area++;
            grid[i][j]=0;
            //Top
            if(i>0)
                dfs(grid, i-1, j);
            //Bottom
            if(i<grid.length-1)
                dfs(grid, i+1, j);
            //Left
            if(j>0)
                dfs(grid, i, j-1);
            //Right
            if(j<grid[0].length-1)
                dfs(grid, i, j+1);
        }
    }
}
```

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.

Find the maximum area of an island in the given 2D array. (If there is no island, the maximum area is 0.)