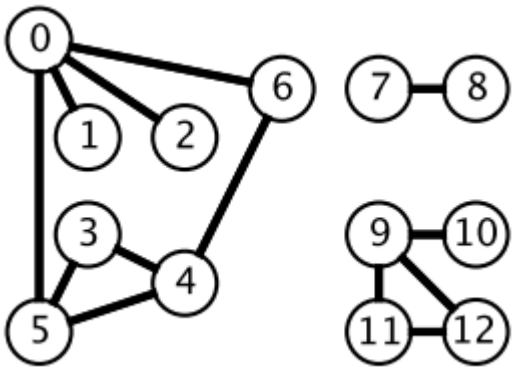
695. 岛屿的最大面积

题目：695. Max Area of Island 语言: python3 英文版链接：https://leetcode.com/problems/max-area-of-island/description/ 中文版链接：https://leetcode-cn.com/problems/max-area-of-island/description/

题目分析

广度优先搜索一层一层遍历，每一层得到的所有新节点，要用队列存储起来以备下一层遍历的时候再遍历。

而深度优先搜索在得到一个新节点时立即对新节点进行遍历：从节点 0 出发开始遍历，得到到新节点 6 时，立马对新节点 6 进行遍历，得到新节点 4；如此反复以这种方式遍历新节点，直到没有新节点了，此时返回。返回到根节点 0 的情况是，继续对根节点 0 进行遍历，得到新节点 2，然后继续以上步骤。



从一个节点出发，使用 DFS 对一个图进行遍历时，能够遍历到的节点都是从初始节点可达的，DFS 常用来求解这种 **可达性** 问题。

在程序实现 DFS 时需要考虑以下问题：

- 栈：用栈来保存当前节点信息，当遍历新节点返回时能够继续遍历当前节点。可以使用递归栈。
- 标记：和 BFS 一样同样需要对已经遍历过的节点进行标记。

对于这道题，我们遍历二维数组的时候，遇到1了，肯定是要检查上下左右是否依然是1（同时注意不要超出边界），如果检查出某一边是1，则还要进一步继续检查它的上下左右是否是1，这说明我们要通过递归来做，遍历时每遇到一个1，就放到递归中去检测并计算岛屿面积。

此外，为了避免循环计算重复的区域，我们要改变已经计算过的岛屿的位置的值，可以从1改成0。

这种递归方式其实就是一种DFS，遇到一个1，则找遍其四周及四周的四周等等，来计算一个岛屿面积，同时改变找过的1的值，避免重复计算。

答案

```python
class Solution:
    def maxAreaOfIsland(self, grid: List[List[int]]) -> int:
        if not grid:
            return 0
        self.r, self.c, max_area = len(grid), len(grid[0]), 0
        for i in range(self.r):
            for j in range(self.c):
                max_area = max(max_area, self.dfs(grid, i, j))
        return max_area

    def dfs(self, grid: List[List[int]], n: int, m: int) -> int:
```

```
if n < 0 or n >= self.r or m < 0 or m >= self.c or grid[n][m] == 0:
    return 0
area, grid[n][m]= 1, 0
area += self.dfs(grid, n - 1, m)
area += self.dfs(grid, n + 1, m)
area += self.dfs(grid, n, m + 1)
area += self.dfs(grid, n, m - 1)
return area
```