**Max area of Island**

**Goal: count the number of adjacent continuous 1s.**

**Algo:**

1. **Visit each position in an array.**
2. **Do a DFS from each position if the value is 1.**
3. **In recursion function you should return count.**
4. **Once the node is visited it should be made 0 so that it wont be considered twice.**
5. **‘break condition is if it is out of boundery or if the value is 0 then 0 will be returned.**
6. **For the node do DFS(Recursion) from all the directions and increment the count and return it.**

**Ref Link: https://www.youtube.com/watch?v=JP39wU1UhRs**

var maxAreaOfIsland = function(grid) {

var max = 0;

var count = 0;

var n = grid.length; // row

var m = grid[0].length; // col

for(var i = 0; i < n; i++) {

for(var j = 0; j < m;j++) {

if(grid[i][j] == 1) {

count = dfs(i , j);

max = Math.max(count,max)

}

}

}

function dfs(row, col) { **// Break condition**

if(row >= n || row < 0 || col >= m || col < 0) { **// Condition to check if the position is inside the boundary of array**

return 0;

}

if(grid[row][col] == 0) { **// Break condition**

return 0;

}

grid[row][col] = 0; // **Assign 0 because that position should not be counted again**

var left = dfs(row - 1, col);

var right = dfs(row + 1, col);

var top = dfs(row, col - 1);

var bottom = dfs(row, col + 1);

return left + right + top + bottom + 1;

}

return max;

};