Matrix DFS

Matrix DFS

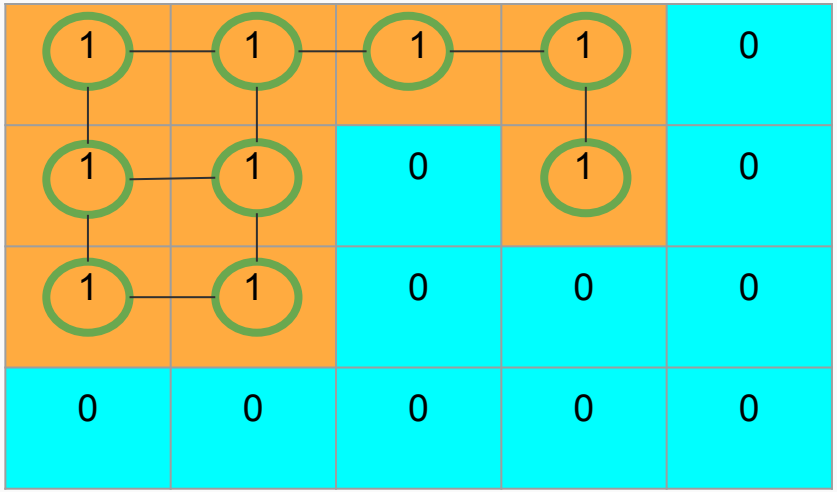
- Matrix DFS is similar to binary tree DFS with a few minor changes
  - Instead of recursively calling on 2 paths (left and right), we must call 4 directions (up, down, left, right)
  - With graphs, we must ensure that we do not re-visit visited nodes.
    - One way we can solve this is by tracking visited nodes in a “visited set”.
  - We must ensure that we do not go out of bounds.
- Demo: Number of Islands

Number of Islands

```
const numIslands = (grid) => {
        let count = 0;
        const visited = new Set();

        for (let row = 0; row < grid.length; row++) {
                for (let col = 0; col < grid[0].length; col++) {
                        if (dfs(grid, row, col, visited)) count++;
                }
        }
        return count;
}


const inBounds = (grid, row, col) => {
        const rowInbounds = 0 <= row && row < grid.length;
        const colInbounds = 0 <= col && col < grid[0].length;
        return rowInbounds && colInbounds;
}
```

```
const dfs = (grid, row, col, visited) => {
        if (!inBounds(grid, row, col)) return false;
        const pos = row + "," + col;
        if (visited.has(pos)) return false;
        if (grid[row][col] === "0") return false;
        visited.add(pos);

        const directions = [[1,0], [0,1], [-1,0], [0,-1]]

        for (let dir of directions) {
                const newRow = row+dir[0];
                const newCol = col+dir[1];
                dfs(grid, newRow, newCol, visited);
        }
        return true;
}
```

Number of Islands



Questions?

Let’s practice!

- Review
  - Max Area of Island
  - Pacific Atlantic Water Flow
- Bonus
  - Word Search