75 Days of Code
Day 48
Problem no : 994
Problem Title :Rotting Oranges
Type : Graph / BFS

You are given an m x n grid where each cell can have one of three values:

0 representing an empty cell,
1 representing a fresh orange, or
2 representing a rotten orange.
Every minute, any fresh orange that is 4-directionally adjacent to a rotten orange becomes rotten.

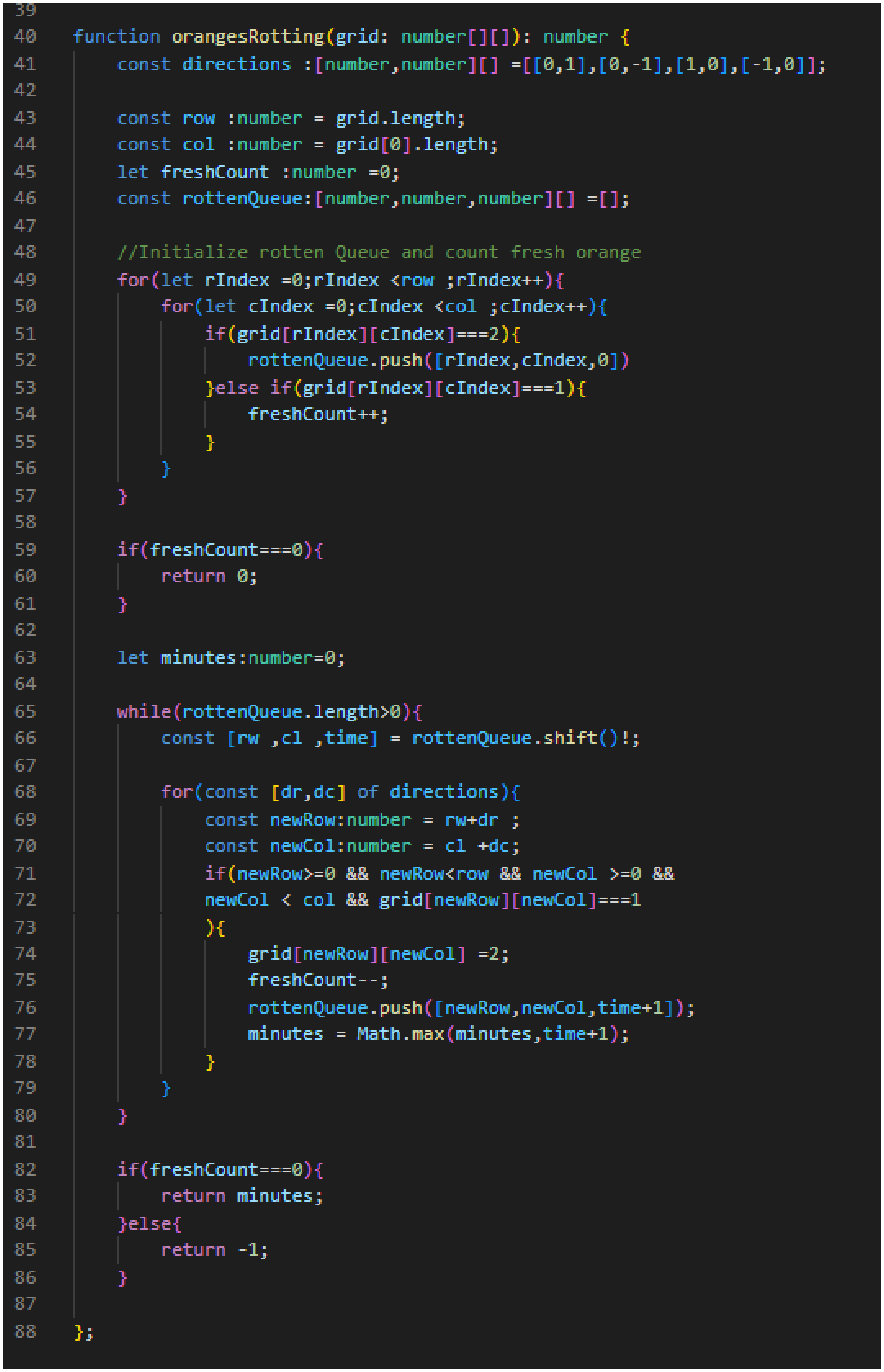
Return the minimum number of minutes that must elapse until no cell has a fresh orange. If this is impossible, return -1.

Example 1:

Input: grid = [[2,1,1],[1,1,0],[0,1,1]]
Output: 4
Example 2:

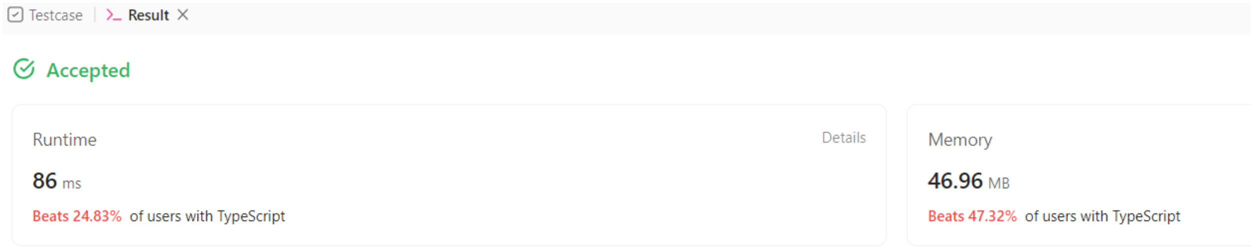
Input: grid = [[2,1,1],[0,1,1],[1,0,1]]
Output: -1
Explanation: The orange in the bottom left corner (row 2, column 0) is never rotten, because rotting only happens 4-directionally.
Example 3:

Input: grid = [[0,2]]
Output: 0
Explanation: Since there are already no fresh oranges at minute 0, the answer is just 0.



```
function orangesRotting(grid: number[][]): number {
    const directions :[number,number][] =[[0,1],[0,-1],[1,0],[-1,0]];

    const row :number = grid.length;
    const col :number = grid[0].length;
    let freshCount :number =0;
    const rottenQueue:[number,number,number][] =[];

    //Initialize rotten Queue and count fresh orange
    for(let rIndex =0;rIndex <row ;rIndex++){
        for(let cIndex =0;cIndex <col ;cIndex++){
            if(grid[rIndex][cIndex]===2){
                rottenQueue.push([rIndex,cIndex,0])
            }else if(grid[rIndex][cIndex]===1){
                freshCount++;
            }
        }
    }

    if(freshCount===0){
        return 0;
    }

    let minutes:number=0;

    while(rottenQueue.length>0){
        const [rw ,cl ,time] = rottenQueue.shift()!;

        for(const [dr,dc] of directions){
            const newRow:number = rw+dr ;
            const newCol:number = cl +dc;
            if(newRow>=0 && newRow<row && newCol >=0 &&
            newCol < col && grid[newRow][newCol]===1
            ){
                grid[newRow][newCol] =2;
                freshCount--;
                rottenQueue.push([newRow,newCol,time+1]);
                minutes = Math.max(minutes,time+1);
            }
        }
    }

    if(freshCount===0){
        return minutes;
    }else{
        return -1;
    }

};
```

Shubham Agrahari



Accepted

Runtime

86 ms

Beats 24.83% of users with TypeScript

Memory

46.96 MB

Beats 47.32% of users with TypeScript

Shubham Agrahari