

1) (10 pts) DSN (Recursive Coding)

Model an area of land as a two dimensional grid of integers, where each integer represents the elevation of that portion of land. Water can only flow from a grid square of higher elevation to lower elevation in one of the four cardinal directions (north, south, east and west). Complete the recursive function below that takes in a 2D array of integers storing the elevation levels of each portion of land, another 2D array of integers (storing 0 or 1 in each entry) representing which grid squares have been flooded with water (1 for flooded, 0 for not flooded), as well as the current row and column value of a grid square that just flooded, and marks the current and all subsequent squares that will get flooded as a result of the water at the given location. Once a square is flooded it remains in that state. An inbounds function and DR,DC arrays are provided for convenience.

```
#define NUMROWS 10
#define NUMCOLS 12

const int DR[] = {-1,0,0,1};
const int DC[] = {0,-1,1,0};

int inbounds(int row, int col);

void floodfill(int grid[][NUMCOLS], int flooded[][NUMCOLS], int row, int col) {

    if ( !inbounds(row, col) ) return;           // Grading: 2 pts

    flooded[ row ][ col ] = 1 ;                // Grading: 1 pt

    for (int i=0; i<4; i++) {

        int nextR = row + DR[i] ;                // Grading: 2 pts

        int nextC = col + DC[i] ;                // Grading: 2 pts

        if ( grid[nextR][nextC] < grid[row][col] ) // Grading: 3 pts

            floodfill(grid, flooded, nextR, nextC);

    }

}

int inbounds(int row, int col) {
    return row >= 0 && row < NUMROWS && col >= 0 && col < NUMCOLS;
}
```

Grading Notes: Give partial credit for slots as necessary, subtract a total of 2 points if rows and columns are switched consistently (the function prototype infers that columns is the second index), don't take off for any extra checks such as seeing if flooded is 0 before doing the recursion...this turns out not to be necessary due to the acyclic structure of this specific problem.