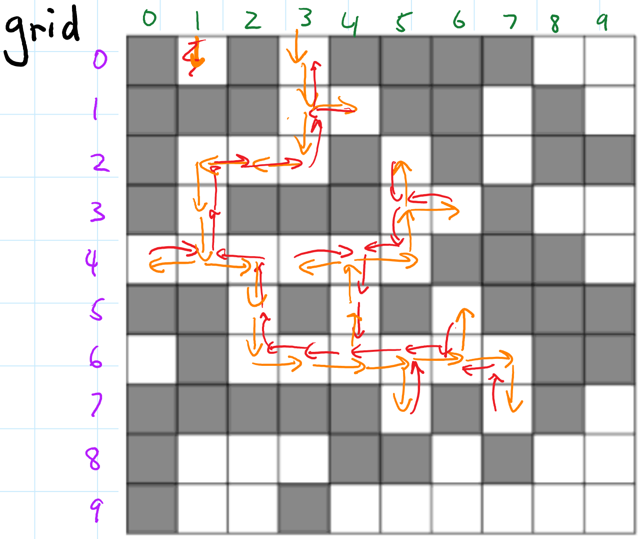
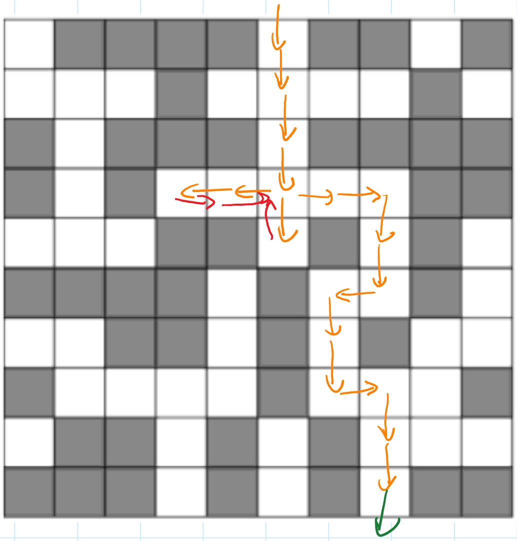
# DETECTING PERCOLATION



grid[row, col]



Depth-First Search (DFS)

ANALOGY: Suppose there is an "agent" in each open square?

When queried, each agents responds as follows:

- If agent is on the bottom row, then say so.
- Otherwise, query the agent below. Is there a path from the square below? If so, then done.
- Otherwise, query the agent at left. Is there a path? If so, then done.
- Otherwise, query the agent at right. Is there a path? If so, then done.
- Otherwise, query the agent above. Is there a path? If so, then done.
- Otherwise, answer "no".

# PSEUDOCODE:

```
                                  matrix of
                                     0,1
def query(row, col, grid, visited):
            indexes of                   another matrix that stores
          square to query                which squares have been
                                                 visited

    #mark row, col as visited

    #is row equal to the bottom row? If so, output "path found"

    #If square below is open and unvisited, then is there
        a path from that square?
    if ______________________________:
        query(row+1, col, grid, visited)

    #If square at left is open and unvisited, then is there
        a path from that square?

    #If square at right is open and unvisited, then is there
        a path from that square?

    #If square above is open and unvisited, then is there
        a path from that square?
```