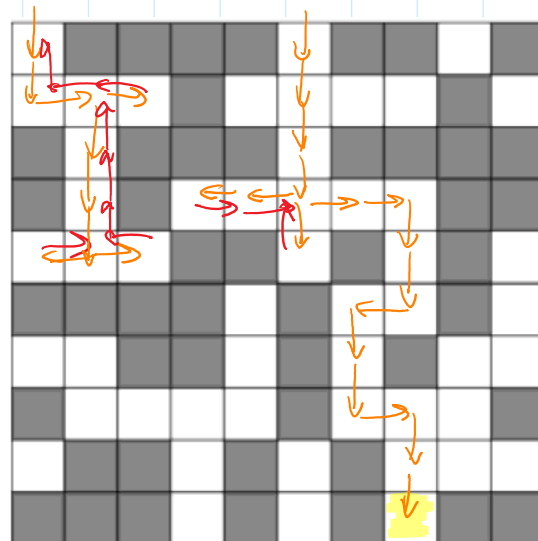
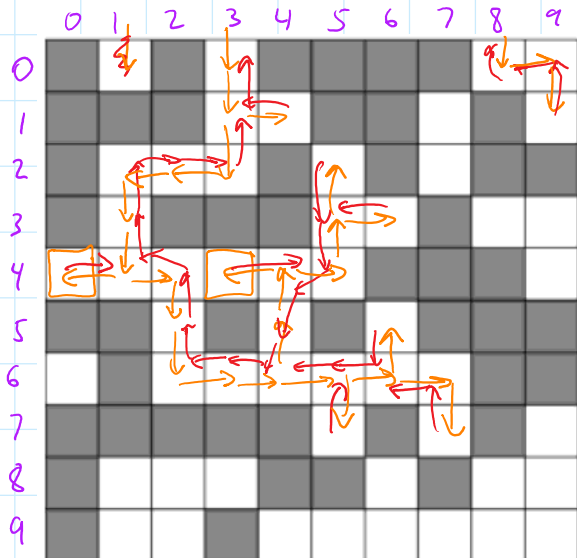


DETECTING PERCOLATION

Down
Left
Right
Up



Done!

Depth-First Search (DFS)

ANALOGY: Imagine there is an "agent" in each square.

When queried, each agent responds as follows:

- If I am on the bottom, respond "path found".
- Otherwise, is there an agent at the square below?
If so, query that agent.
- Otherwise, is there an agent at the square to left?
If so, query that agent.
- Otherwise, is there an agent at the square at right?
If so, query that agent.
- Otherwise, is there an agent at the square above?
If so, query that agent.
- Otherwise, report "no".

PSEUDOCODE:

```
def query(row, col, grid, visited):  
    # mark row, col as visited
```

0/1
matrix

matrix that remembers
which agents have
been queried

indexes of agent
to query

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

```
# if square below is open and unvisited, then query it  
if _____:  
    query(row+1, col, grid, visited)
```

```
# if square at left is open and unvisited, then query it
```

```
# if square at right is open and unvisited, then query it
```

```
# if square above is open and unvisited, then query it
```