

# MATH 242: Friday, April 24

## TODAY:

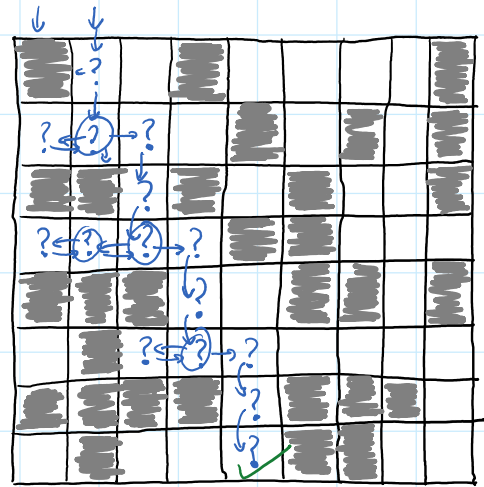
- Applications of percolation theory
- Examine ways of creating the random number grid in Python
- Algorithm for detecting percolation

## EXAMPLE:

STRATEGY: Each time I visit an open square, ask "Am I on the bottom row?"

- If yes, then I'm done.
- If no, then ask, "Is there a path from any adjacent, unvisited open square to the bottom row?"

left,  
down,  
right,  
up



percolation  
path found!

## PSEUDOCODE:

```
def findPath(row, col, visited):
```

```
# if row is bottom row, then done ←
```

```
# mark square (row, col) as visited
```

```
# otherwise:
```

```
# if cell to left is open and unvisited:
```

```
findPath(row, col-1)
```

```
# if cell below is open and unvisited:
```

```
findPath(row+1, col)
```

RECURSIVE  
FUNCTION  
(it calls itself)

# if cell to right is open and unvisited:

findPath(row, col+1)

# if cell above is open and unvisited:

findPath(row-1, col)