

Recursive Depth-First Search for a Percolation Path

Function `query` detects whether there exists a percolation path from `(row, col)` to the bottom of `grid`.

`query(grid, row, col, visited):`

mark `row, col` as visited

print `row, col`

if `row` is the bottom row, then print "path found"

otherwise, look for a path from an adjacent square

if square at right is open and unvisited:

`query(grid, row, col+1, visited)`

if square at left is open and unvisited:

`query(grid, row, col-1, visited)`

if square below is open and unvisited:

`query(grid, row+1, col, visited)`

if square above is open and unvisited:

`query(grid, row-1, col, visited)`

Function `findPercolation` starts the search at each open square in the top row.

`findPercolation(grid):`

initialize visited array

for each open square in the top row:

`query(grid, 0, col, visited)`

print "done"