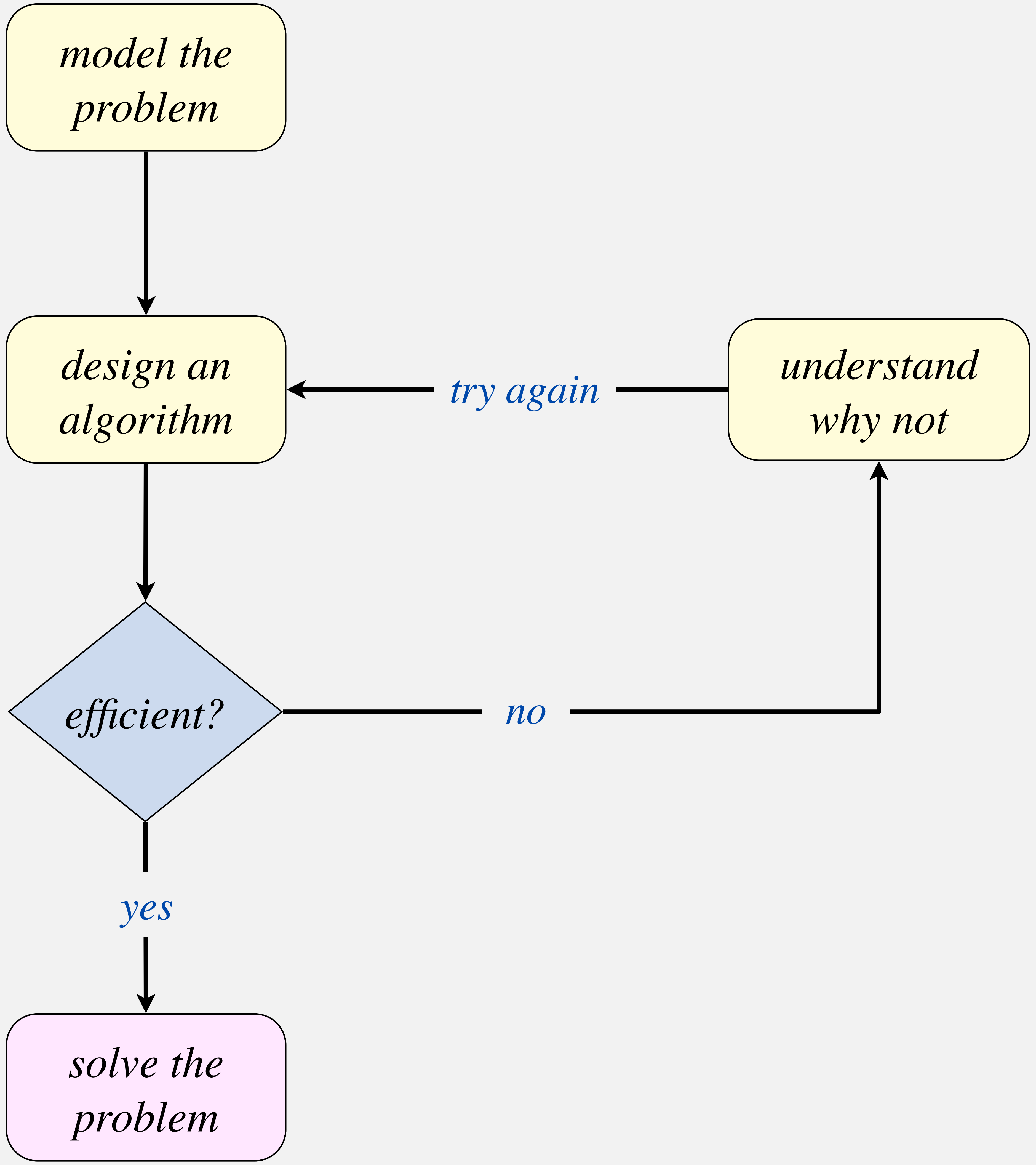
https://algs4.cs.princeton.edu

Percolation

Last updated on 9/1/21 8:31 PM

Subtext of today's lecture (and this course)

Steps to develop a usable algorithm to solve a computational problem.



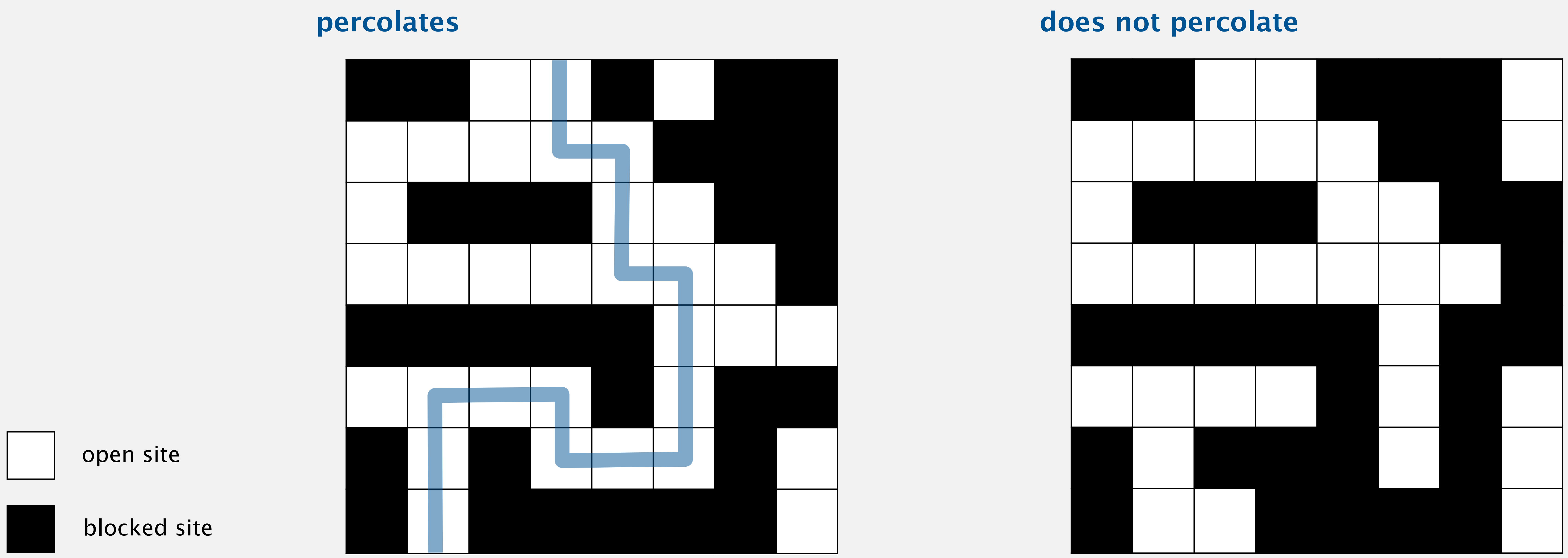
Percolation

An abstract model for many physical systems:

- n-by-n grid of sites.
- Each site is open with probability p.
- System **percolates** if there is a path of open sites between top and bottom.

left, right, up, down

Intuition. Pour liquid on top of a porous material. Will the liquid reach the bottom?

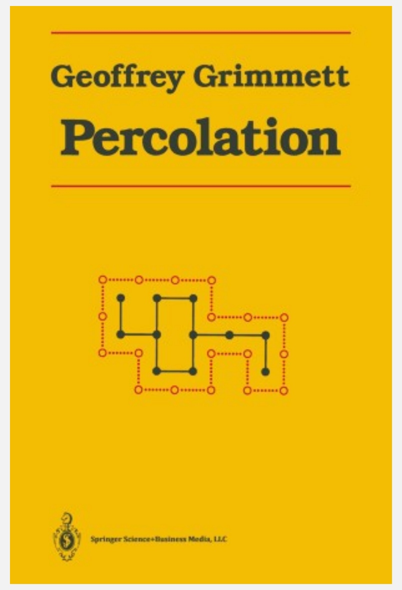


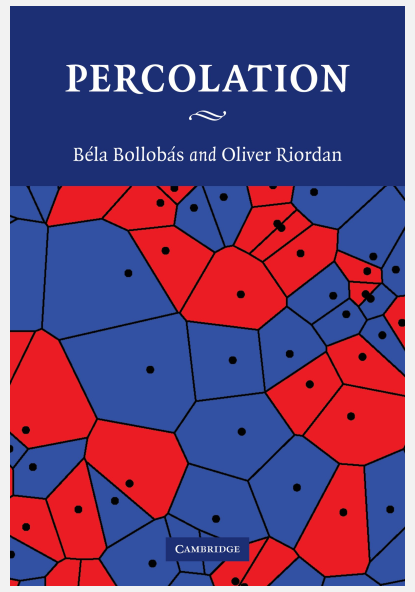
Percolation

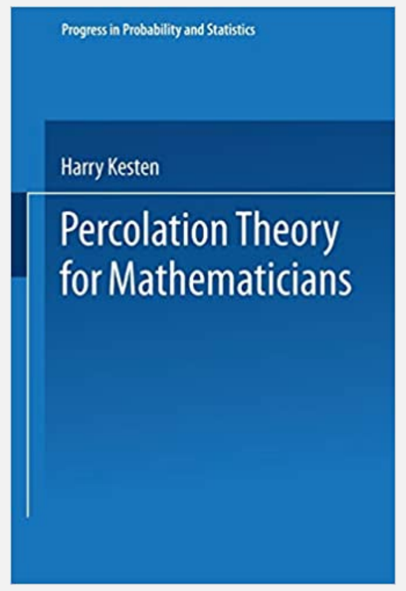
An abstract model for many physical systems:

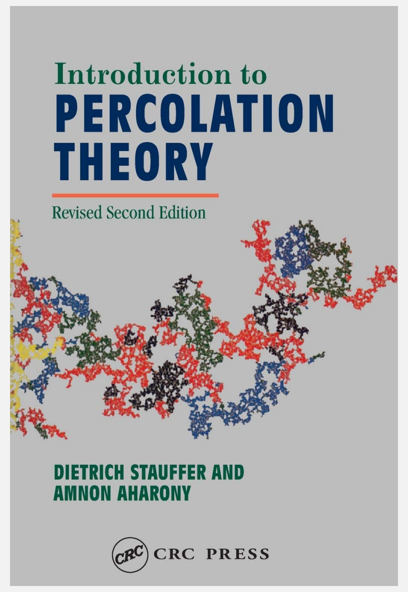
- n-by-n grid of sites.
- Each site is open with probability p.
- System percolates if there is a path of open sites between top and bottom.

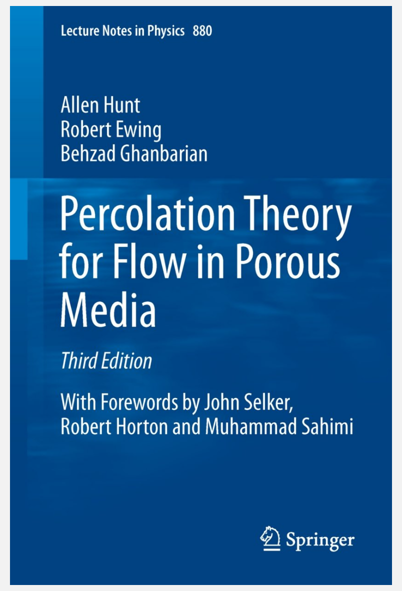
system	open site	blocked site	percolates
geological	cavity	rock	porous
electrical	metallic	dielectric	conductor
ecological	tree	no tree	wildfire
epidemiological	not socially distanced	socially distanced	pandemic

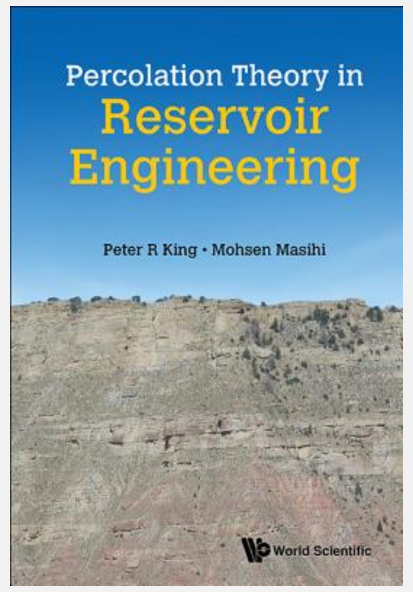


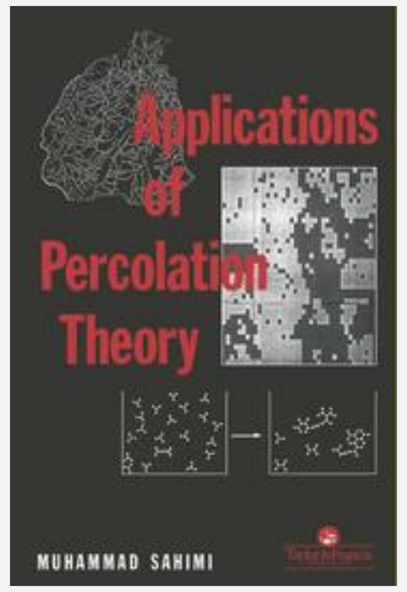


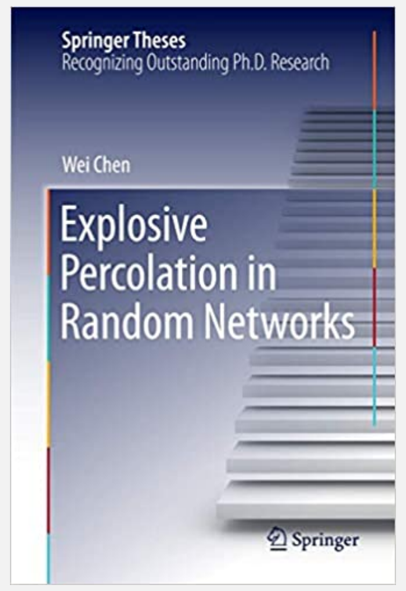












Percolation quiz 1

Which of the following systems percolate?

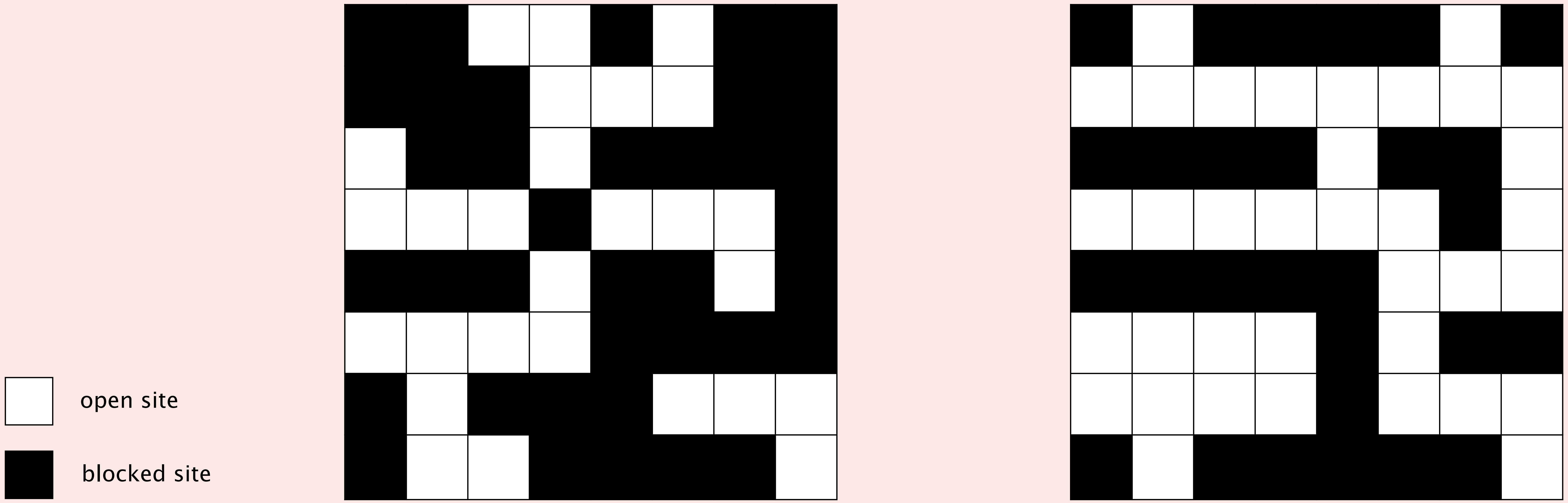
A. A only.

B. B only.

C. Both A and B.

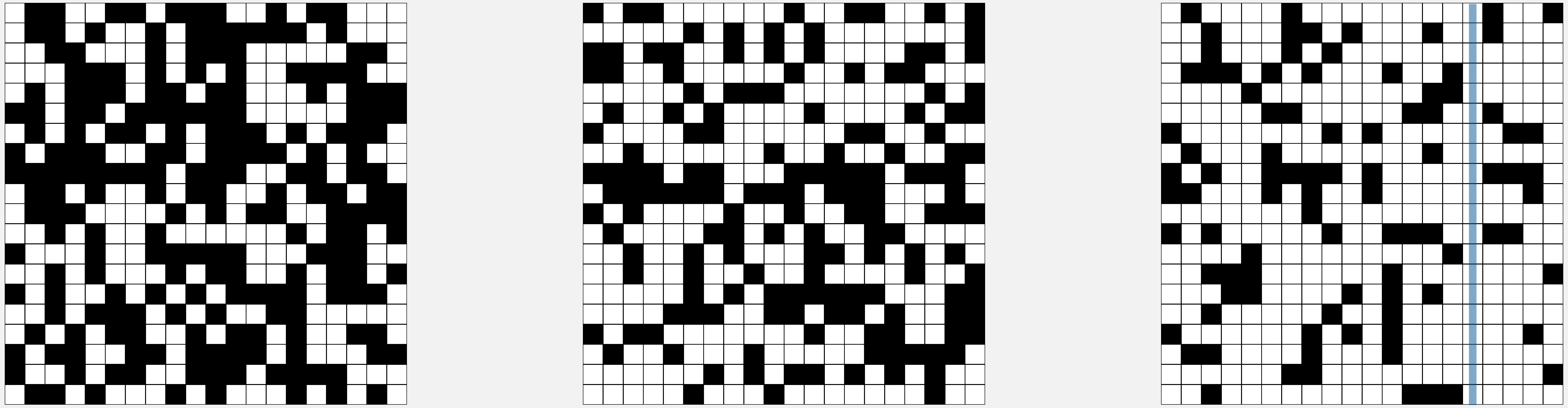
D. Neither A nor B.

A. **B.**



Likelihood that system percolates

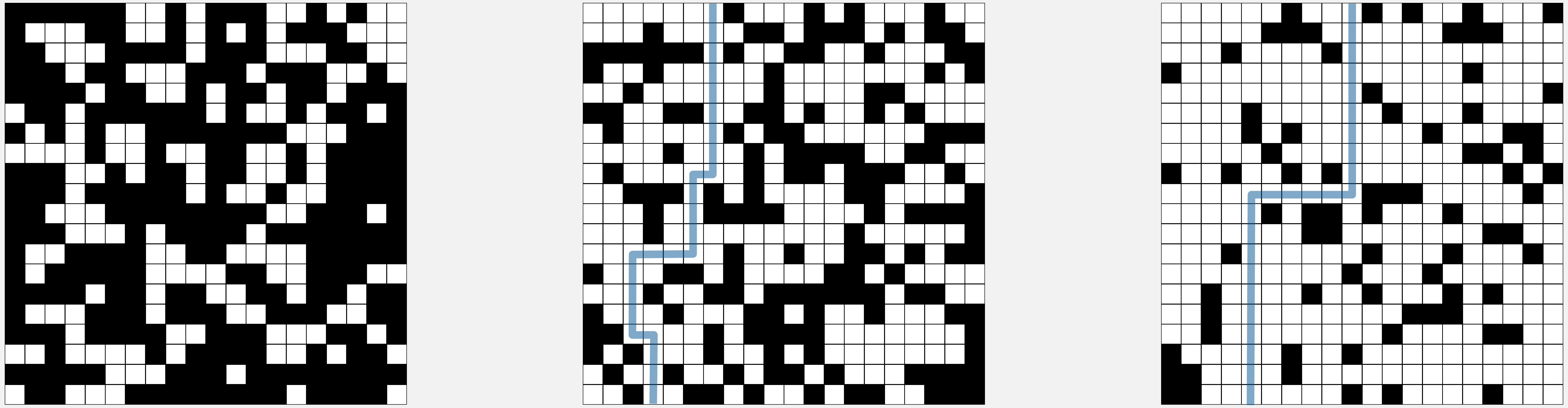
Depends on grid size n and site vacancy probability p.



p low (0.4)
does not percolate

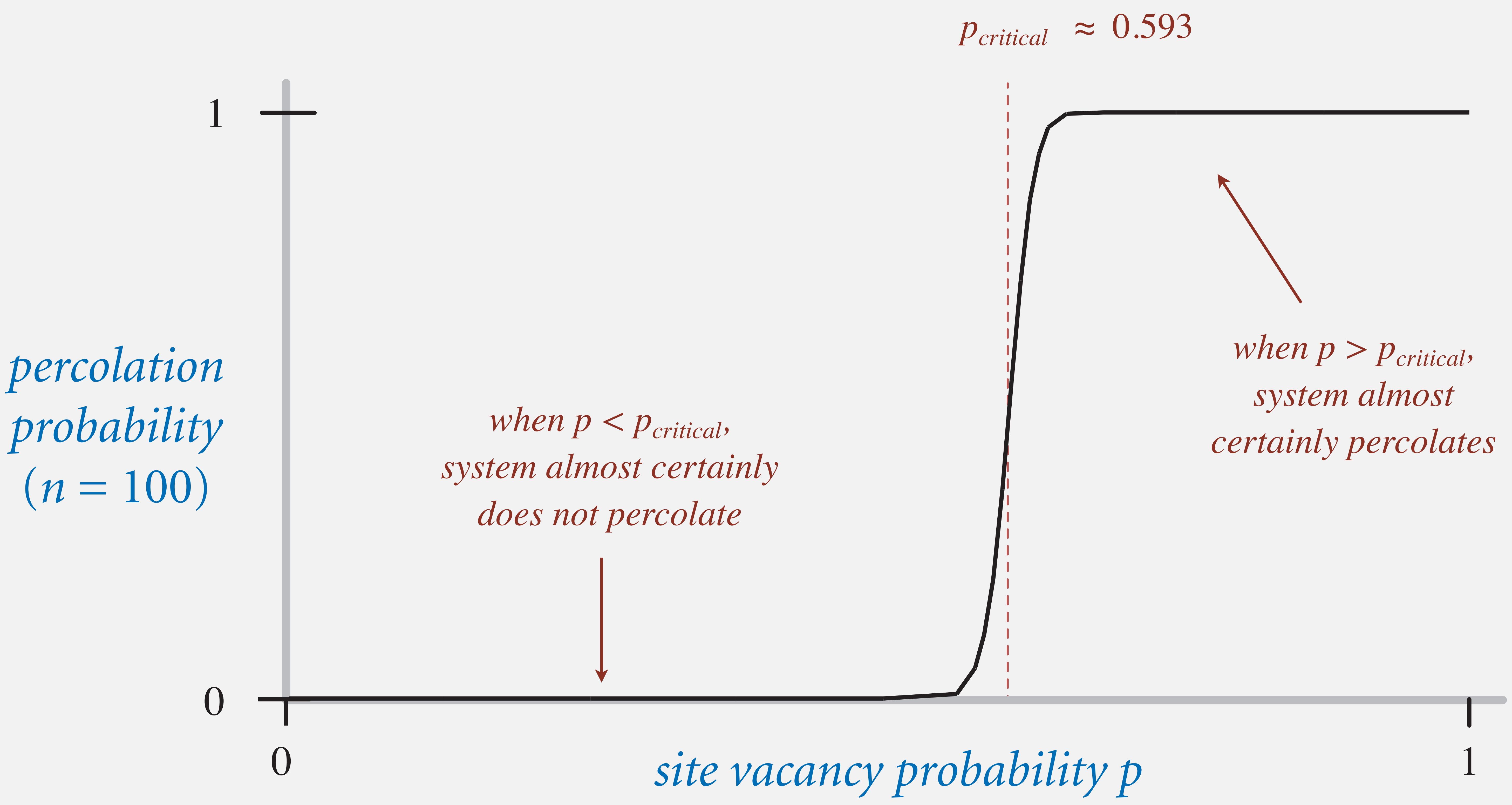
p medium (0.6)
percolates?

p high (0.8)
percolates



Percolation phase transition

Phase transition. When n is large, probability theory guarantees a sharp threshold $p_{critical}$.



Q. What is the value of $p_{critical}$?

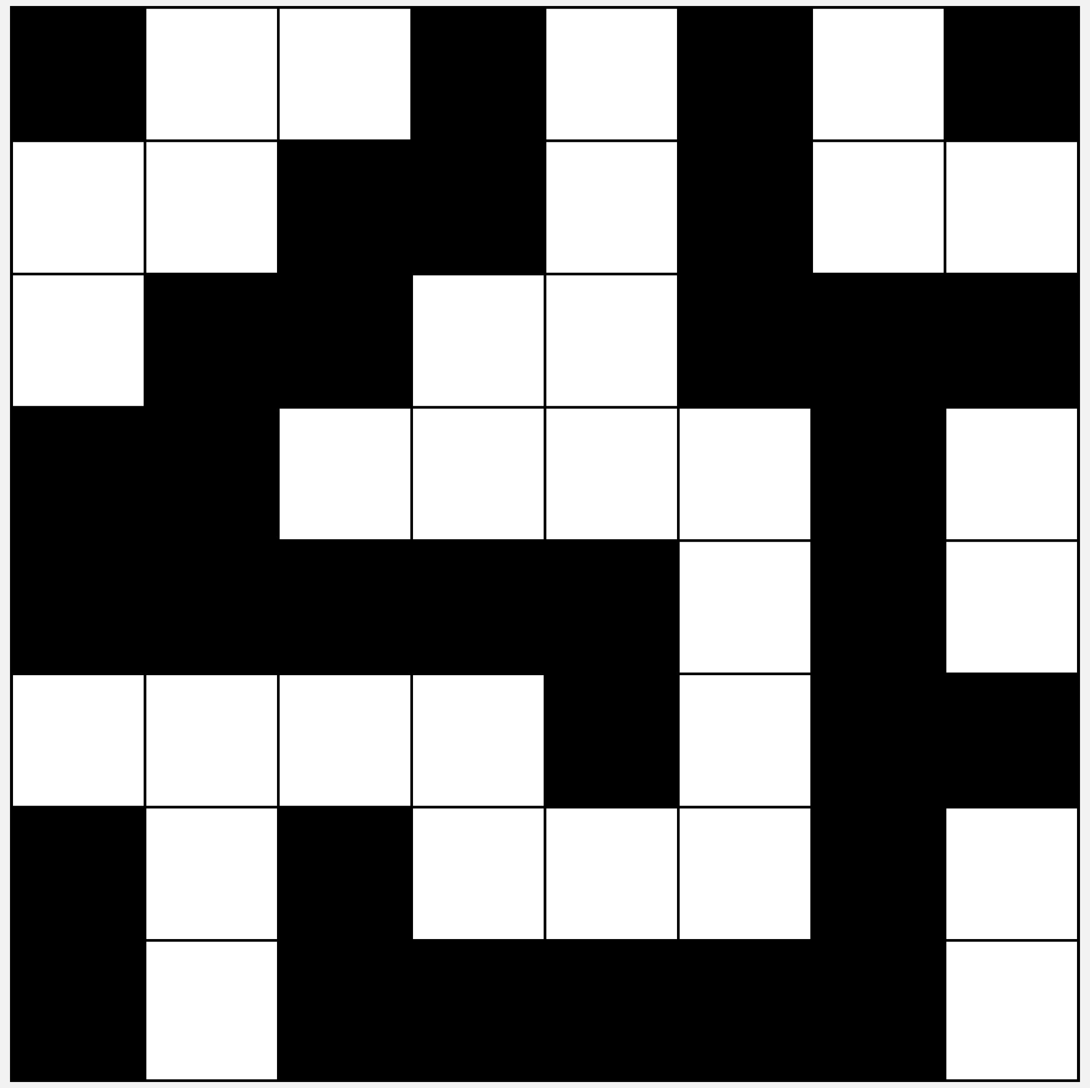
A. No analytic solution known. $\Rightarrow$ Estimate via computational experiments.

Monte Carlo simulation

A computational experiment.

- Initialize all sites in an n-by-n grid to blocked.
- Open sites one at a time, uniformly at random, until system percolates.
- Fraction of sites opened = estimate of percolation threshold $p_{critical}$.

Monte Carlo simulation. Repeat experiment many times to get accurate estimate.

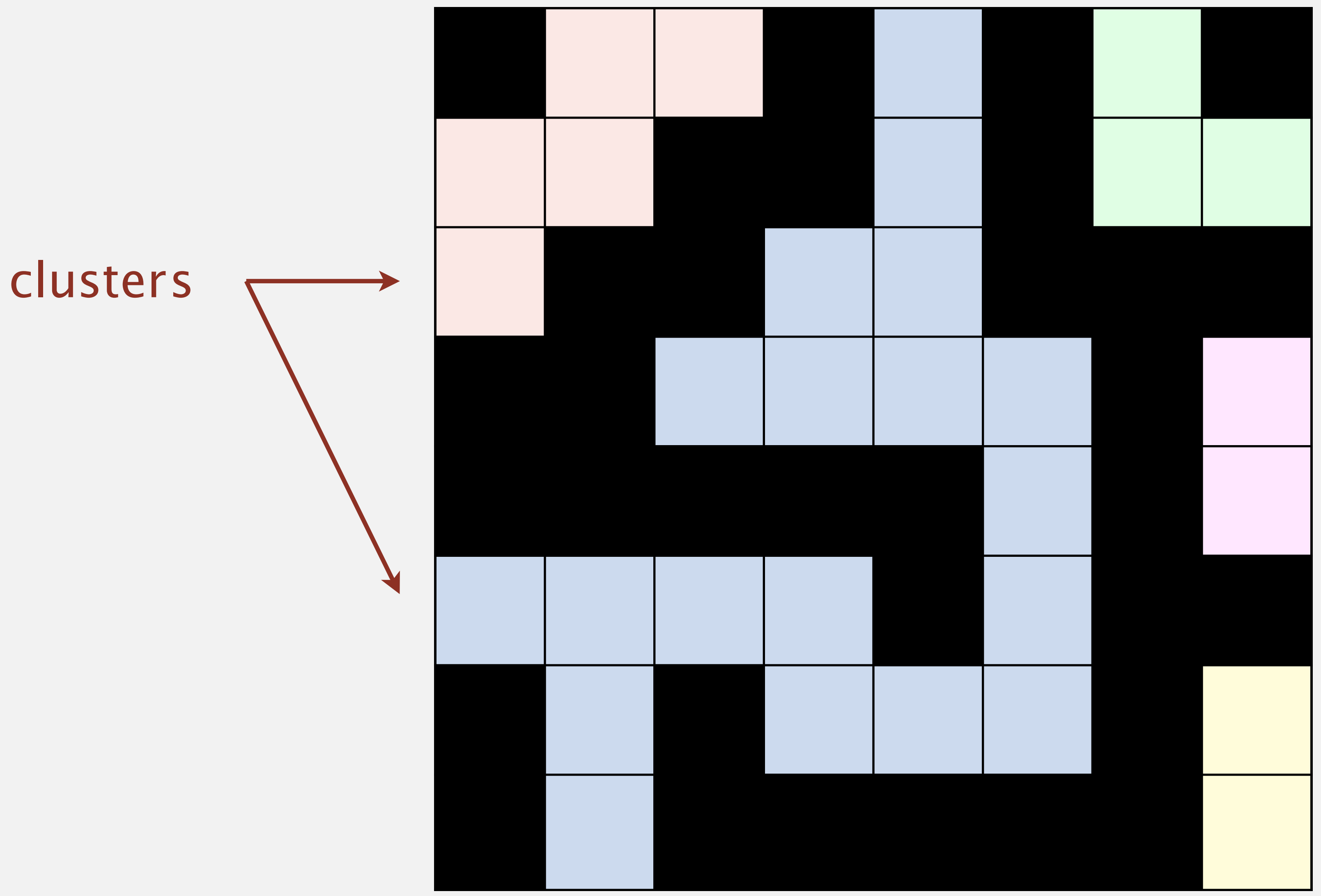


estimate = 31 / 64

Clusters

Cluster. Maximal set of open sites connected via path of open sites.

Observation 1. Percolates if any site in top row is in same cluster as any site in bottom row.

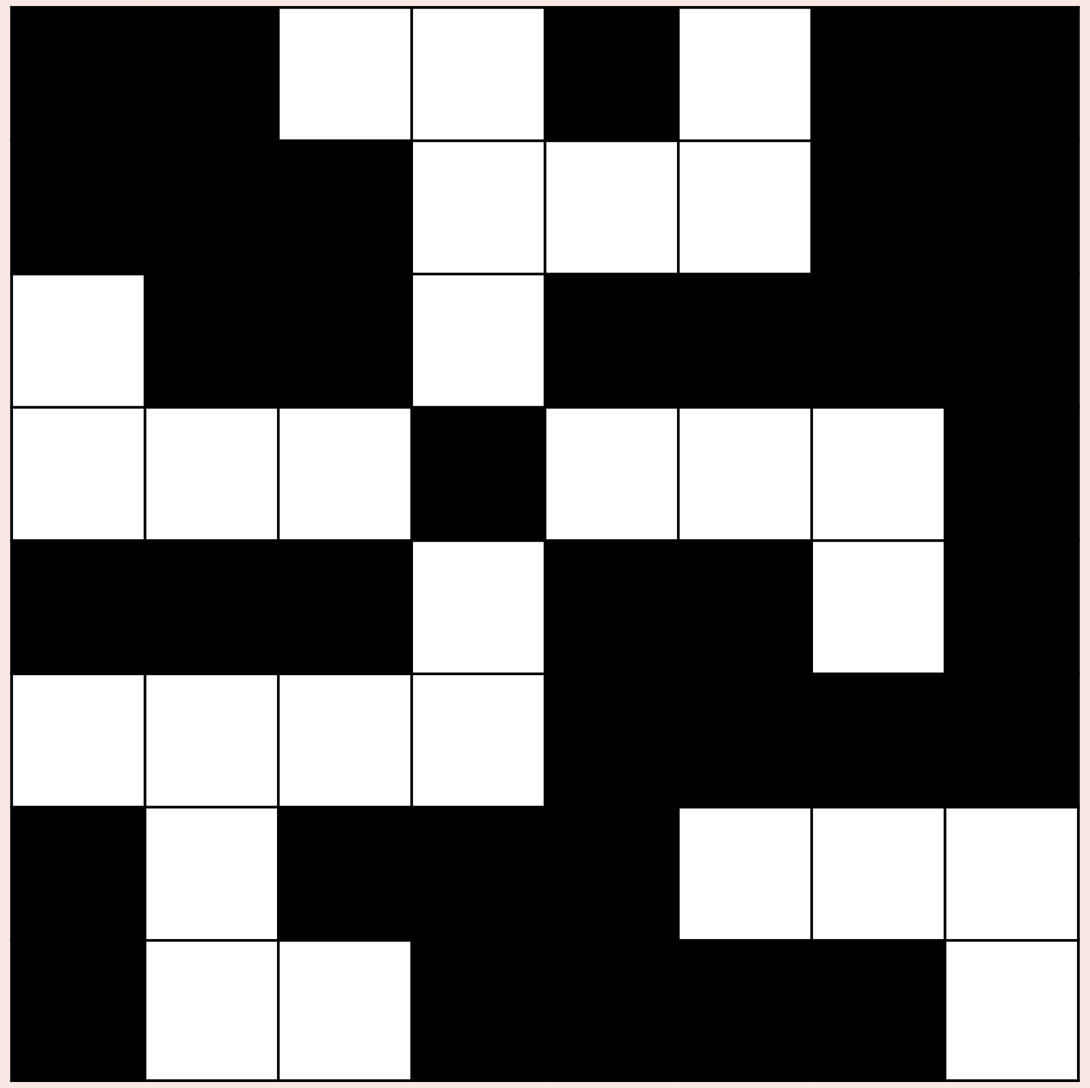


5 clusters

Percolation quiz 2

How many clusters?

- **A.** 2
- **B.** 3
- **C.** 4
- **D.** 5



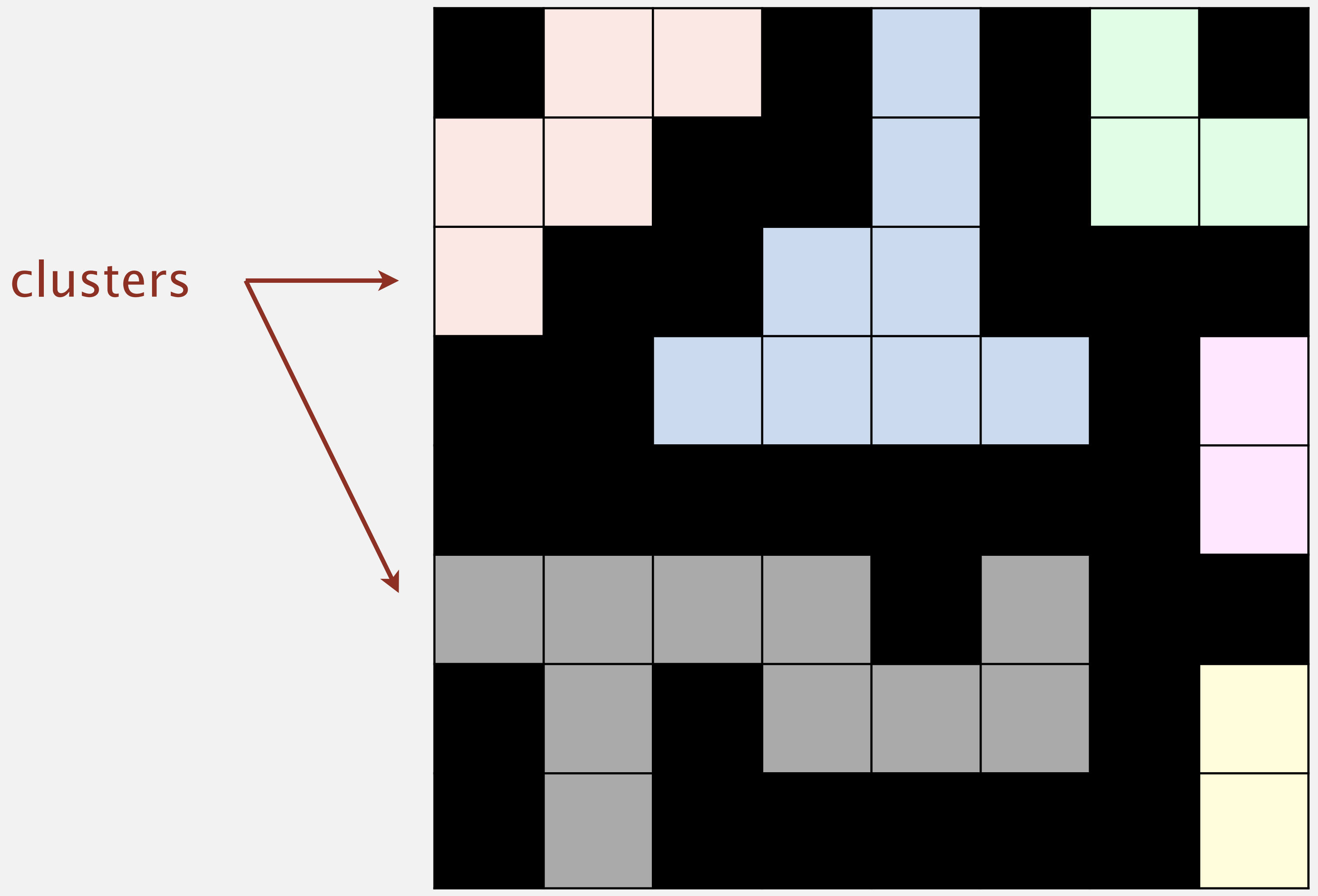
Computing clusters

Cluster. Maximal set of open sites connected via path of open sites.

Q. How to compute clusters?

A. Could use depth-first search.

[but takes time proportional to n^2 and must do after opening each site]



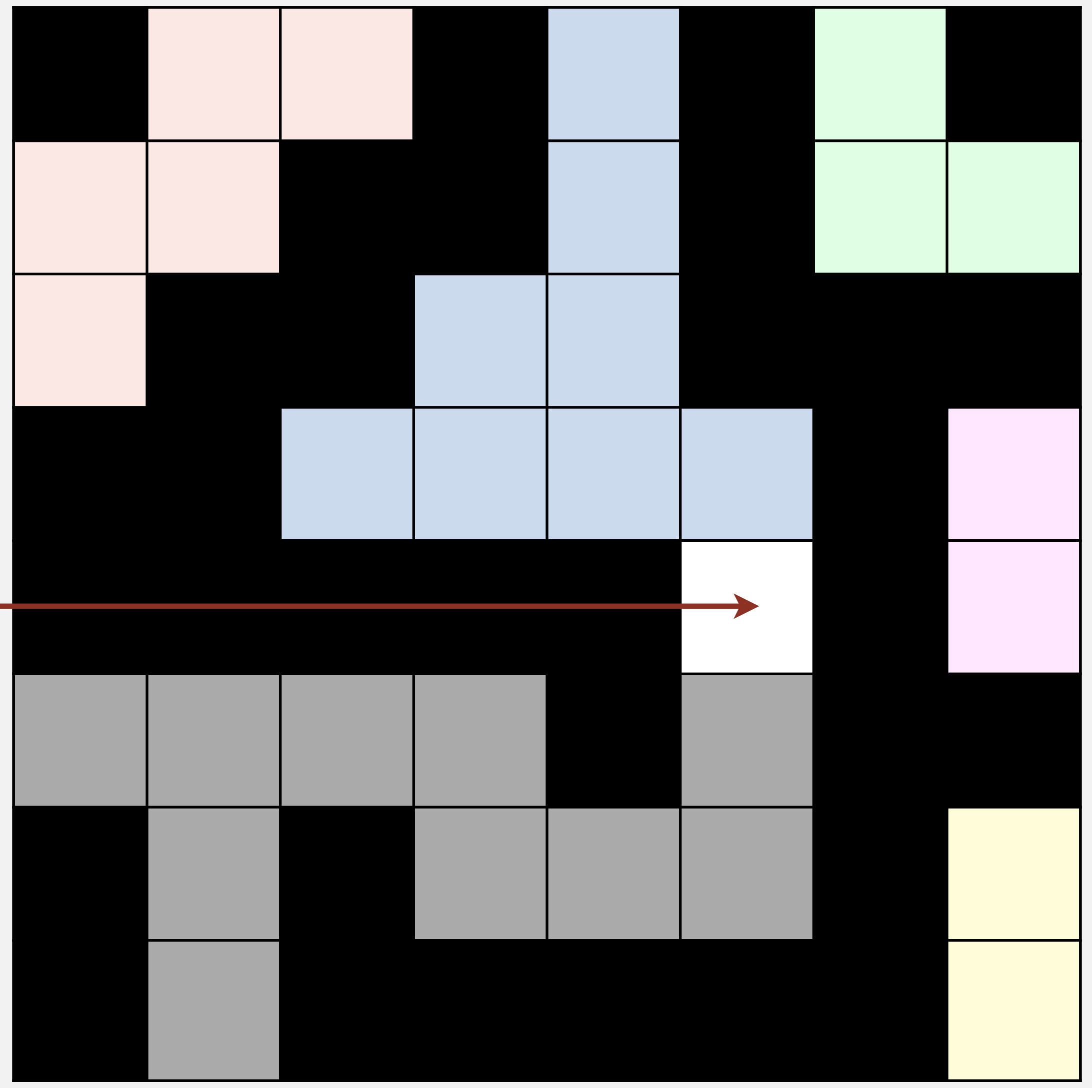
Maintaining the clusters

Observation 2. When opening a site, few clusters change.

Goal. Devise a data structure to maintain clusters when opening a site.

- MAKE: create a new cluster with one site.
- UNION: merge two clusters.
- FIND: in which cluster is a given site?

new site to open



Percolation quiz 3

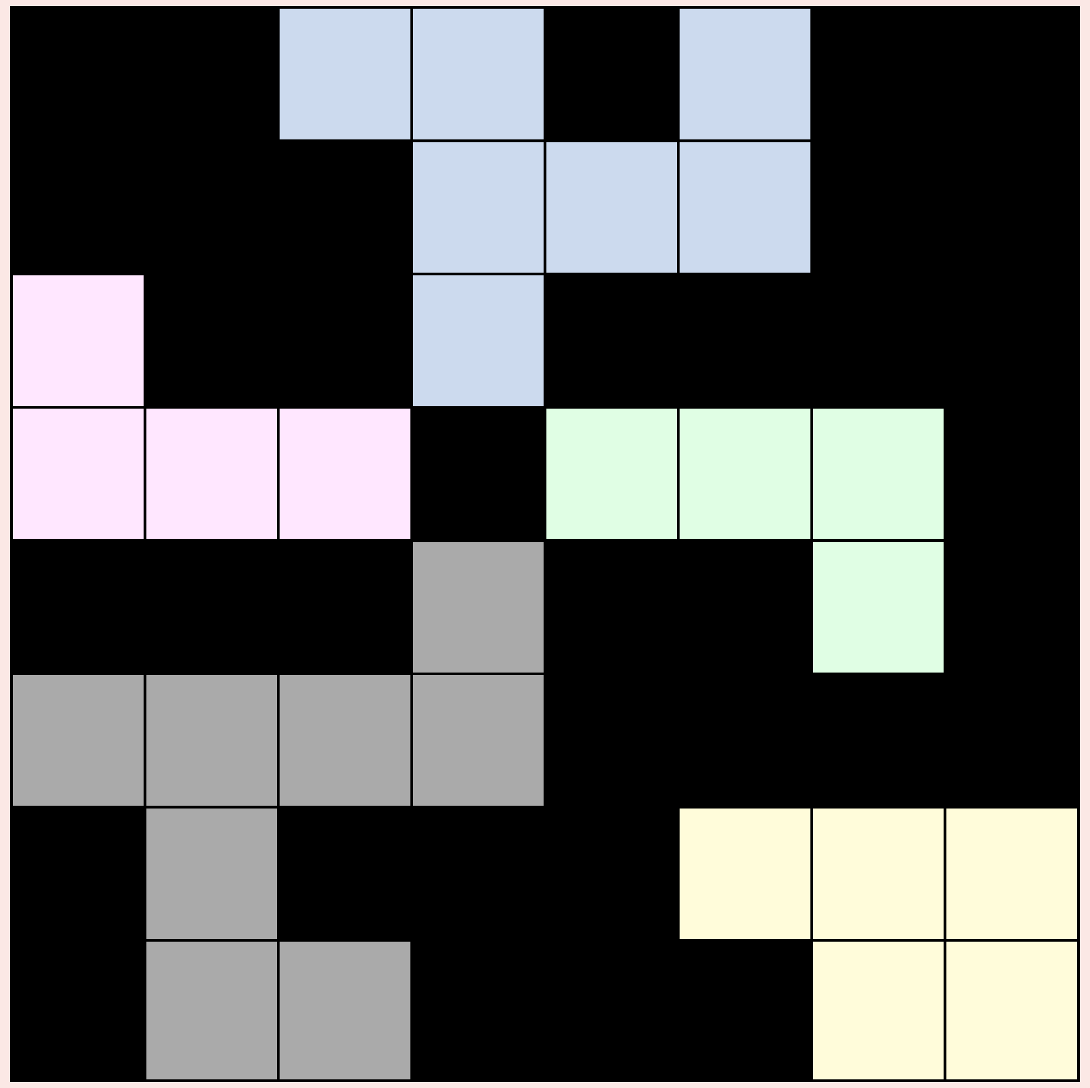
When opening a site in an n-by-n system, what is the max number of UNION operations that might need to be performed to update the clusters?

A. 1

B. 2

C. 3

D. 4



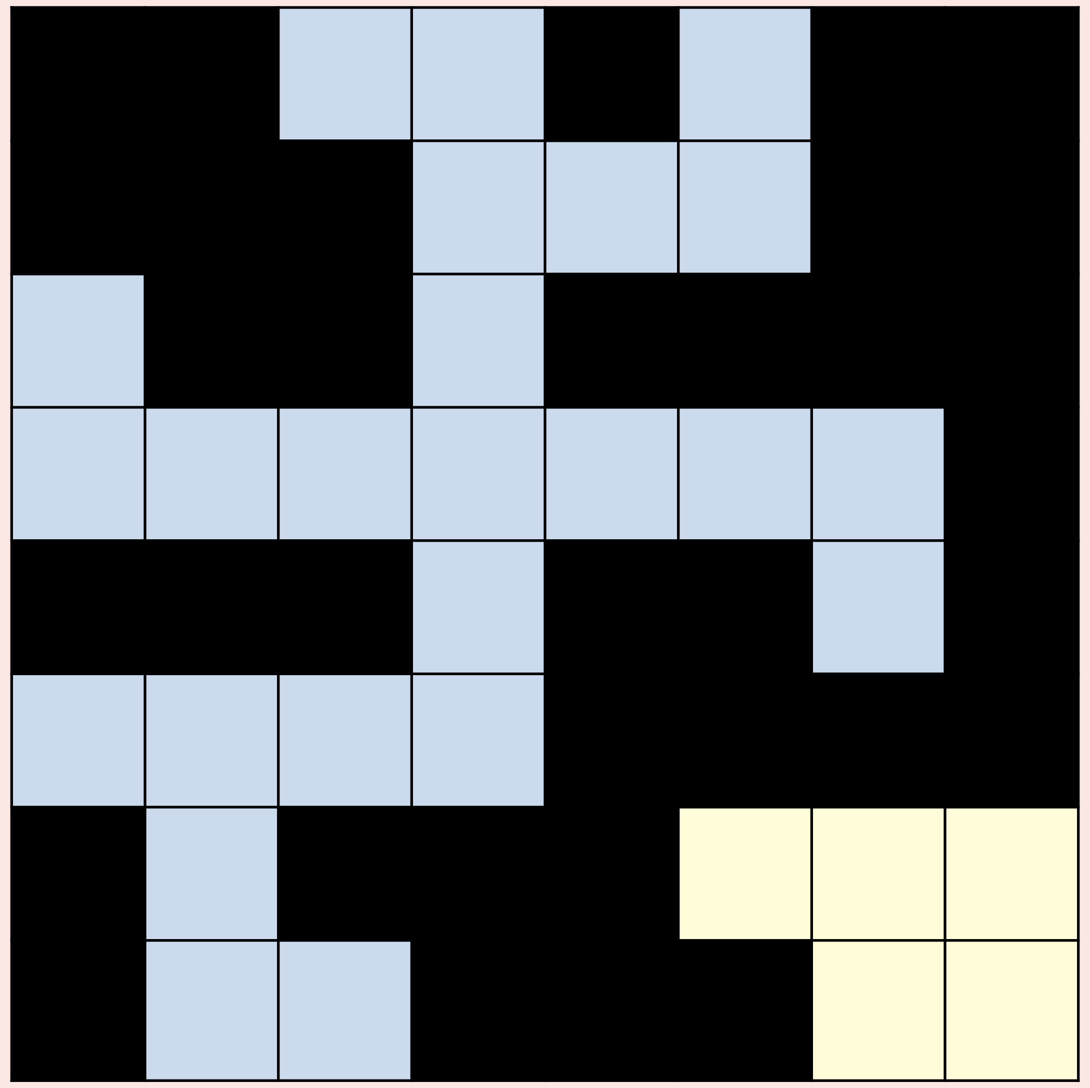
After opening a new site in an n-by-n system, what is the max number of FIND operations that might need to be performed to check whether the system percolates?

A. 2

B. $2n$

C. $2n^2$

D. $2n^3$

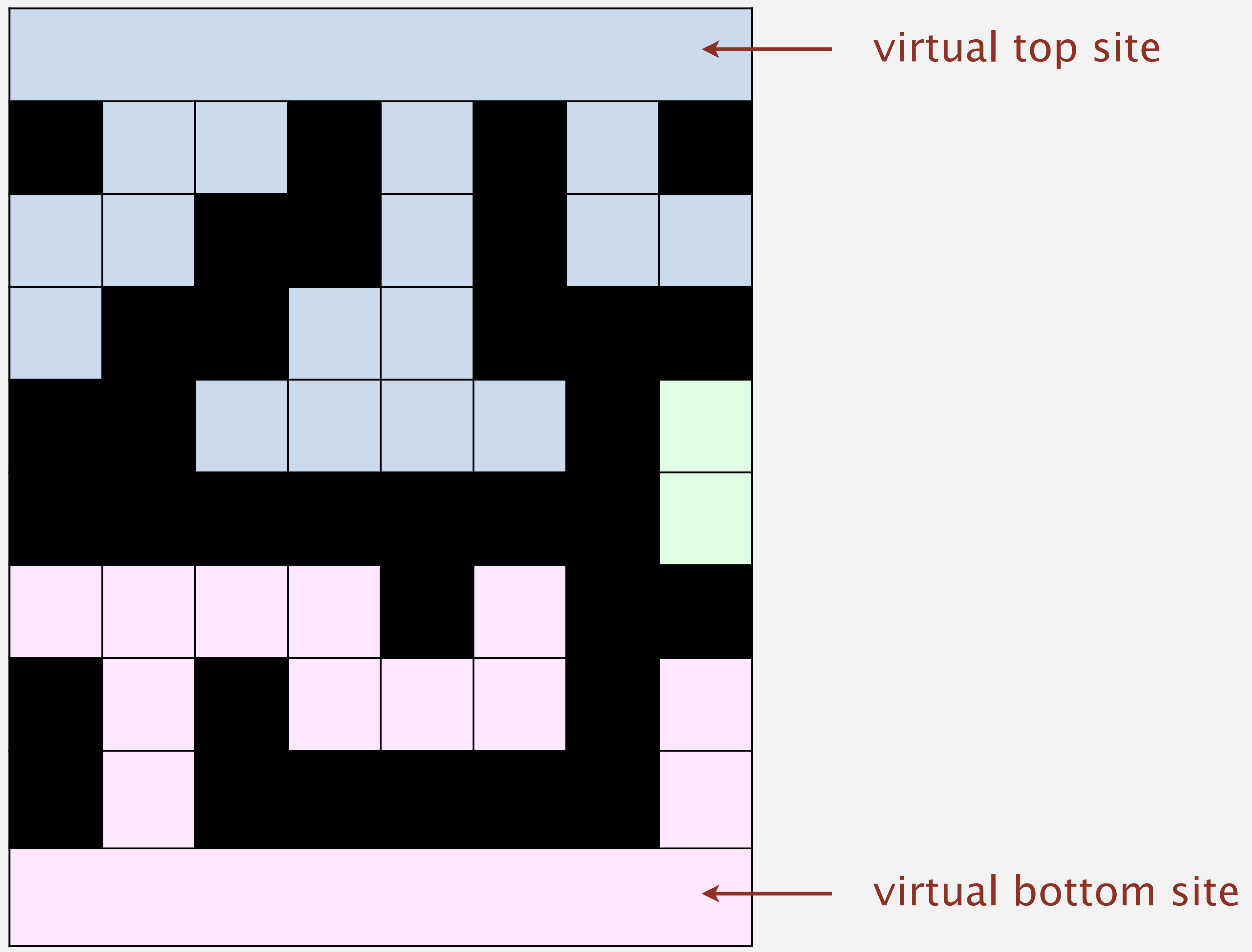


A clever optimization

Key idea. Create a "virtual" top site and a "virtual" bottom site.

Observation. System percolates if virtual top and bottom sites are in same cluster.

Impact. Now, two calls to FIND suffices to detect percolation.



Union–find data structure

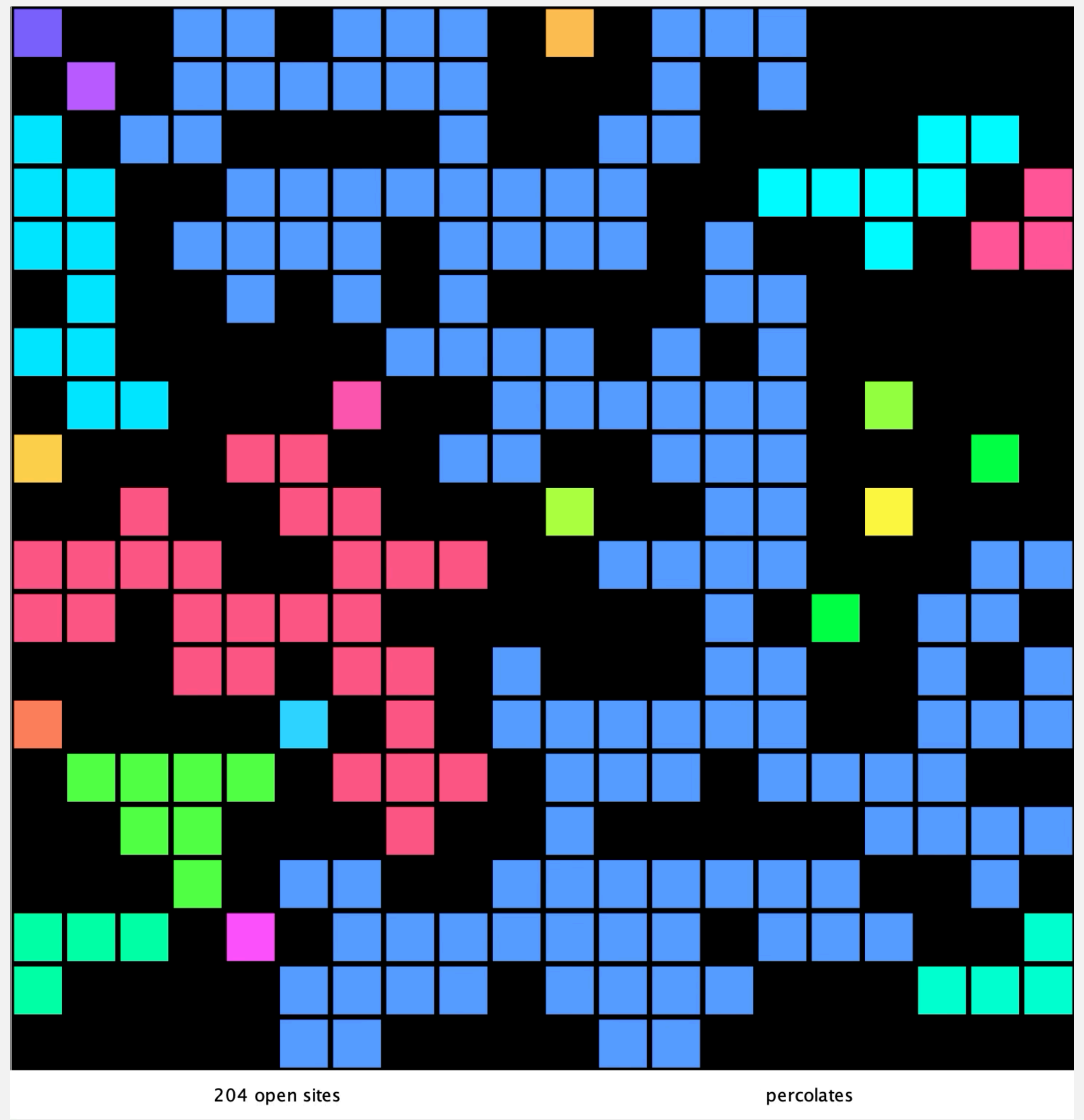
Summary. To perform an experiment, open sites one at a time and maintain clusters.

Opening a site involves (a constant number of) two core operations:

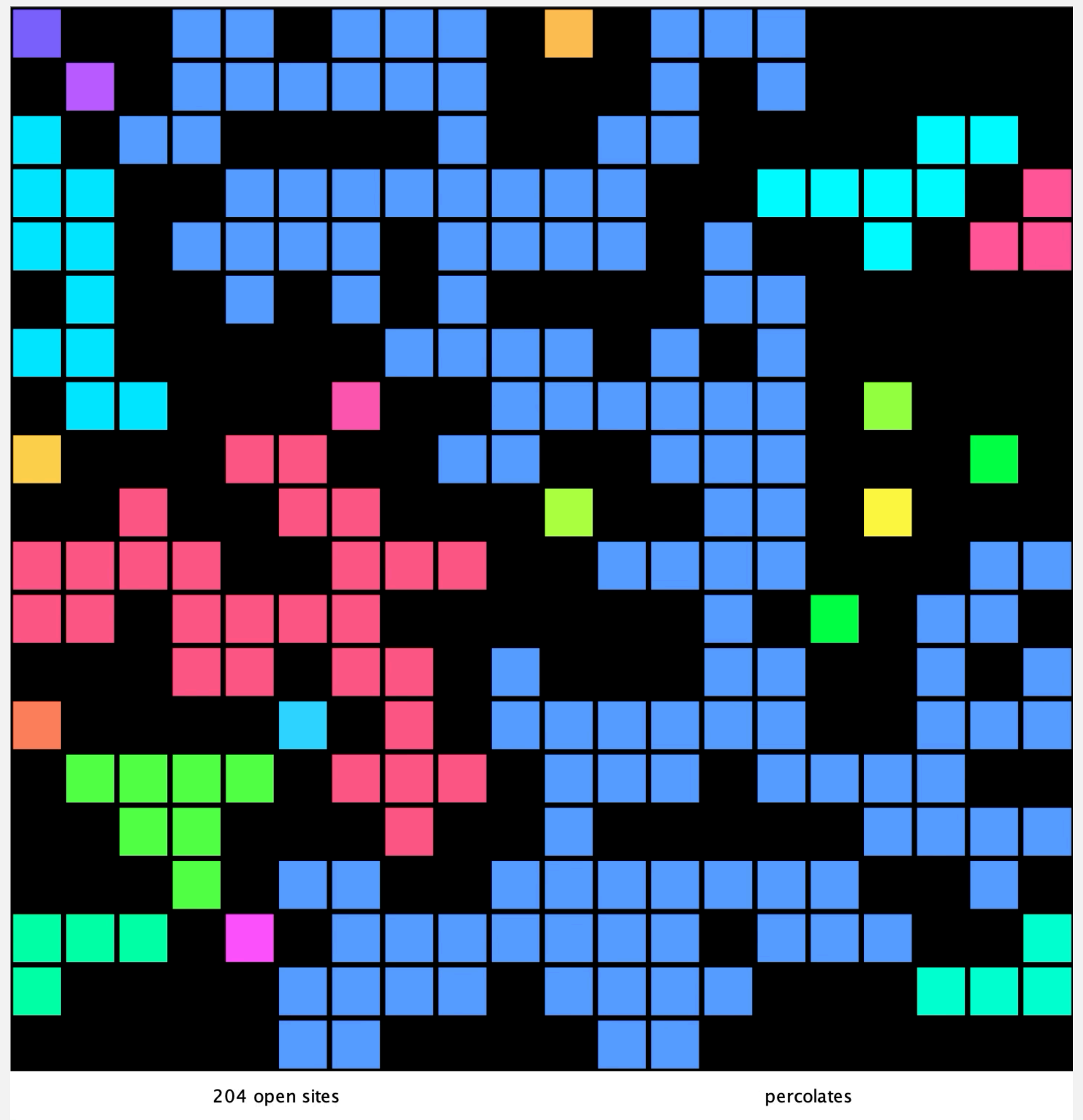
- UNION: merge two clusters.
- FIND: in which cluster is a given site?

Next lecture. Design an efficient data structure to support UNION and FIND.

Impact. Ingenious algorithms enable scientific progress (and much more)!



brute force



ingenious