CS200 - Programming Assignment 2: Percolation

Motivation

Percolation refers to the movement of a fluid through a porous material, but there are many systems that are mathematically similar, and percolation theory is the study of the general principles behind percolation. The following are examples of percolation: water flowing through ground coffee beans to make coffee, a crack developing in wood or stone, decay in a tooth, and disease spreading through a population. We are generally interested in the probability that a fluid will easily flow through the system. There is often a parameter than controls how flow happens in the system, and it is important to determine the *critical threshold*, which is the value of the parameter when flow starts to occur.

Guidelines

Please read and abide by the honor code guidelines in the syllabus.

Please read the rubric so you know how you will be graded. For example, turning in a program that compiles and runs without errors but does nothing will earn you more points than a program that is close to working but does not compile or contains errors on running.

You may use any functions, methods, or packages that you find useful, and you may use code snippets that you find on, for example, Stack Overflow (as long as you properly cite your reference).

Put a multi-line comment at the beginning of your program. It should contain:

- Your name
- "Programming Assignment 2"
- The name of anyone you worked with and the nature of your collaboration
- Sample output from your program
- The amount of time (approximately) that you spent on this assignment

Assignment

Consider an undirected, unweighted graph on n vertices (with no self loops), where each edge is included in the graph with probability p. We say there is flow on the graph when the graph is connected. A graph is connected if, starting at vertex 1, you can run a graph search algorithm and find all other vertices in the graph. Let E be the event that includes all elements of the sample space that have flow. Write a program in java or python that takes as input n and p, and determines the probability of E. For 1 bonus point, additionally create a variant that takes as input n and returns the value of p that causes $P(E)$ to equal 1/2 - this is the *critical threshold.*

Some questions you should think about before starting:

- What is the sample space? What is its size?
- If a graph has b edges, what is the probability of that graph occurring?
- You should calculate $P(E)$ using one of the following formulas: $P(E) = |E|/|S|$ or $P(E) = \sum_{G \in E} prob(G)$. Which one? Why?
- Did you carefully read the entire paragraph titled "Assignment?" Go back and read it again.

Here are some hints:

- Even though this is a question about probability, your code should NOT contain any randomness.
- In the last programming assignment, you had to iterate through all possible values of true and false in a truth table. In this assignment, you'll have to iterate through all possible graphs...there are some similarities in these problems.
- Start early!
- Ask questions early!