Percolation Project

Math 242

due Friday, April 30

We have been examining the 2D percolation problem in which we have a $n \times n$ grid of squares. Each square is either "open" with probability p or "closed" with probability $1 - p$. We say that percolation occurs if there is a path of open squares from any square in the top row to any square in the bottom row of the grid.

Your Tasks

Investigate the following questions:

1. **How does the probability of percolation depend on p when n is fixed?** Let $n = 10$ and find the probability of percolation for various values of p. Make a plot showing how the probability of percolation depends on p when $n = 10$. Then repeat this for $n = 20$, $n = 30$, $n = 40$, and $n = 50$. You may try additional values of n if you like. What do you observe?

2. **How does the probability of percolation depend on n when p is fixed?** Let $p = 0.5$ and find the probability of percolation for various values of n. Make a plot showing how the probability of percolation depends on n when $p = 0.5$. Then repeat this for $p = 0.55$, $p = 0.6$, $p = 0.65$, and $p = 0.7$. You may try additional values of p if you like. What do you observe?

Note: When working with larger grid sizes, you may experience a "maximum recursion depth exceeded" error. If so, increase Python's recursion depth limit by running the following code:

```
import sys
sys.setrecursionlimit(4000)
```

If necessary, replace 4000 with a larger number in the code above.

Your Report

Turn in your investigation as a Python Colab notebook. Make sure you clearly describe how the probability of percolation depends on p and n. Be as specific as you can, and include computations to support your answers.

As usual, submit code that runs and explain what your code does. Your goal should be to communicate your work to another person (e.g., another student at your level who is not in this course).

Grading Rubric

Your notebook will be graded on a scale of 0 to 16 points. The following rubric gives characteristics of notebooks that will merit sample point totals. (Interpolate the following for point totals that are not divisible by 4.)

16 points. Problems and goals are clearly stated, including relevant definitions or parameters. Computations are complete; code runs and is clearly explained. Conclusions are clearly stated and backed up by sufficient computational evidence. Limitations of the methodology, extensions for future work, and conjectures are discussed. Notebook is well-formatted and easy to read.

12 points. Problems and goals are stated well, though relevant definitions or parameters may be missing. Computations are mostly complete; code runs, but explanation is weak. Conclusions are unclear or not well justified. Insufficient discussion of limitations, extensions, and conjectures.

8 points. Statement of problem or goal is unclear. Computations are incomplete; explanation is ambiguous. Code may produce errors when run. Conclusions are possibly correct, but not justified. Little or no discussion of limitations, extensions, or conjectures. Notebook is difficult to read.

4 points. Serious misunderstanding of the problem or goal. Computation is inadequate for the task at hand. Work is not clearly explained. No discussion of limitations, extensions, or conjectures. Notebook is difficult to read.

0 points. Notebook is not turned in.