1. (3 + 3) The 1-d percolation problem:

Consider an infinite 1-d lattice forest with a tree present at any site with probability \( p \).

- Find the distribution of forest sizes as a function of \( p \). Do this by moving along the 1-d world and figuring out the probability that any forest you enter will extend for a total length \( \ell \).
- Find \( p_c \), the critical probability for which a giant component exists. 
  
  \text{Hint: One way to find critical points is to determine when certain average quantities explode. Compute } \langle \ell \rangle \text{ and find } p \text{ such that this expression goes boom (if it does).}

2. Show analytically that the critical probability for site percolation on a triangular lattice is \( p_c = 1/2 \).

   \text{Hint—Real-space renormalization gets it done.:}  
   \text{http://www.youtube.com/v/J1kbU5U7QqU?rel=0}

3. (3 + 3)

   Coding, it’s what’s for breakfast:

   (a) Percolation in two dimensions (2-d) provides a classic, nutritious example of a phase transition.
Your mission, whether or not you choose to accept it, is to code up and analyse the $L$ by $L$ square lattice percolation model for varying $L$.

Take $L = 20, 50, 100, 200, 500, \text{ and } 1000$. 

(Go higher if you feel $L = 1000$ is for mere mortals.)

(Go lower if your code explodes.)

Let’s continue with the tree obsession. A site has a tree with probability $p$, and a sheep grazing on what’s left of a tree with probability $1 - p$.

Forests are defined as any connected component of trees bordered by sheep, where connections are possible with a site’s four nearest neighbors on a lattice.

Do not bagelize (or doughnutize) the landscape (no periodic boundary conditions—boundaries are boundaries).

(Note: this set up is called site percolation. Bond percolation is the alternate case when all links between neighboring sites exist with probability $p$.)

Steps:

i. For each $L$, run $N_{\text{tests}} = 100$ tests for occupation probability $p$ moving from 0 to 1 in increments of $10^{-2}$. (As for $L$, use a smaller increment if that’s just how you do things.)

ii. Determine the fractional size of the largest connected forest for each of the $N_{\text{tests}}$, and find the average of these, $S_{\text{avg}}$.

iii. On a single figure, for each $L$, plot the average $S_{\text{avg}}$ as a function of $p$.

(b) Comment on how $S_{\text{avg}}(p; N)$ changes as a function of $L$ and estimate the critical probability $p_c$ (the percolation threshold).

Helpful reuse of code (intended for black and white image analysis): You can use Matlab’s bwconncomp to find the sizes of components. Very nice.

4. (3 + 3)

(a) Using your model from the previous question and your estimate of $p_c$, plot the distribution of forest sizes for $p \approx p_c$ for the largest $L$ your code and psychological makeup can withstand. (You can average the distribution over separate simulations.)

Comment on what kind of distribution you find.

(b) Repeat the above for $p = p_c/2$ and $p = p_c + (1 - p_c)/2$, i.e., well below and well above $p_c$.

Produce plots for both cases, and again, comment on what you find.