Introduction to Complexity

Homework 1

Graded Part: This part you will submit on the course webpage, by doing the exercises, and then clicking on the "Submit Homework 1" link in Unit 1.10, and filling out the answers. The first three graded questions will require you to download and run the Ant2 model (downloadable from the Course Materials page).

1. In the Ant2 model, with `population-size` = 100, `max-step-size` = 10, and `max-turn-angle` = 60 run the model several times (by clicking on `setup` and `go`) to estimate the average number of ticks (time steps) needed to gather all the food. On average, about how many ticks does it take for the ants to gather all the food?

   A. Less than 100

   B. 100 – 200

   C. 200 – 600

   D. 600 – 1000

   E. More than 1000

2. Do the same thing as in problem 1, but this time change `population` to 50, leaving `max-step-size` = 10, and `max-turn-angle` = 60. On average, about how many ticks does it take for the ants to gather all the food? (Run the model several times, clicking `setup` and `go` each time, and get an approximate average number of ticks.)

   A. Less than 100

   B. 100 – 200

   C. 200 – 600

   D. 600 – 1000

E. More than 1000

3. Now leaving `population` = 50 and `max-step-size` = 10, change `max-turn-angle` to 10. On average, about how many ticks does it take for the ants to gather all the food?

A. Less than 100

B. 100 – 200

C. 200 – 600

D. 600 – 1000

E. More than 1000

4. Suppose the NetLogo procedure below is part of a model's code. What value would `bad-coin-flip?` produce if used somewhere else in the code? (Hint, recall our discussion of Ant1.nlogo.)

```
to-report bad-coin-flip?
  report random 1 = 0
end
```

A. True or false at random

B. Always true

C. Always false

D. An error message

E. None of the above

Ungraded Part: This part is **optional** and ungraded; you won't turn it in. The ungraded assignment is meant for people who want to go further in learning how to create models in NetLogo! This unit's ungraded homework will involve modifying the Ant2.nlogo model's code. You can choose whether you want to do the Beginner, Intermediate, or Advanced options, depending on your level of familiarity and comfort with NetLogo. Feel free to discuss this part of the homework with anyone, or to ask questions about it on the Class Forum.

To learn NetLogo's programming language, like any other language, you'll have to try unfamiliar things and make lots of mistakes! Don't worry, NetLogo won't break if you write something it doesn't understand – instead, it will just give an error message explaining (as best it can) what the problem is.

If you get stuck, or aren't sure what a command does, try using it from the Command Center (see the link on the Course Materials page on How to Use the Command Center), and don't forget to use the Netlogo Dictionary (http://ccl.northwestern.edu/netlogo/docs/dictionary.html) to look up the meanings of keywords.

Beginner Option

1. Modify Ant2.nlogo to change the ants' size to 2.

2. Next modify Ant2.nlogo to **get rid of** the command that labels ants with a number (representing the amount of food they have eaten)

3. Finally modify Ant2.nlogo that make the ants change color depending on how much food they have eaten. That is, if they have eaten more than 50 food patches, they should turn blue, and if they have eaten more than100 food patches they should turn yellow. Hint: Use a statement like

```
if (food-eaten > 50) [set color blue]
```

It's up to you to figure out where to put this, and then how to make them turn yellow if they've eaten more than 100 food patches.

Intermediate Option:

Implement the following: The 9 center patches form the ant nest. Ants wander around as in the Ant2 model, but when an ant finds a patch of food it collects it and returns to the nest before it wanders out again.

Advanced Option

Implement the following, which is similar to the Ants model from the Model Library: Same as the Intermediate Option, but when returning to the nest, the ant leaves a pheromone trail—that is, the patches that it traverses each gain a unit of pheromone. The pheromone evaporates over time—that is, at every time step each patch with pheromone has a probability of losing its pheromone. If a wandering ant encounters a patch with pheromone, it follows the trail as long as it can sense pheromone.

Experiment with this model to see if adding the pheromone mechanism speeds up the ants' process of gathering all the food.