1. HOMEWORK 5 REVIEW.
   a. Volunteers on the whiteboard.
   b. Function that calculates the multiplication of the numbers between \(a\) and \(b\).
      i. While loop.
      ii. For loop.
      iii. Recursive.
   c. Function that checks whether a word is an anagram.
      i. While loop.
      ii. For loop.
      iii. Recursive.

2. DRUNK BABY TURTLES, and how to get them back on their feet.
   a. When drawing the tree, the key is to make sure you give back the turtle right where you got it. Otherwise, the turtle spins out of control.
   b. This is particularly important if you are using random variables. You need to keep track of them within the tree algorithm.
   c. Show example.

3. CUSTOM DATA STRUCTURES.
   a. When solving problems in the real world, or when modeling the real world, we often have situations where it is useful to represent data structures that are slightly more complex than just a value, just a list, or just a dictionary, and where the values have very particular meaning - a one to one relationship with things in the real world.
      i. Example: Point.
         1. An \(x\) and a \(y\) coordinate.
         2. We can use lists or dictionaries to represent a point.
         3. We can create operations on this data structure.
            a. Add two points to create another point.
               i. We can extend this in two ways: Add a list of points together.
               ii. Or add n-dimensional points together.
            b. Find the midpoint of the two points.
               i. We can extend this function in the same two ways again.
               ii. Find the Euclidean distance between two points.
      ii. Example: Time.
         1. An hour, a minute, and a second.
         2. We can again use a list or a dictionary.
         3. We can create operations on this data:
            a. Add two times together, and return the resulting time.
   b. Sometimes it is useful to have composite data structures. Data structures that build upon other data structures.
      i. Example: Rectangle.
         1. There are two ways of representing a rectangle.
            a. Two points.
            b. A width, a length, and the bottom left corner.
         2. We can again use lists or dictionaries to represent a rectangle.
         3. If the data structure is made of sub-structures, we can re-use the operators of the substructures to figure out things about the structure.
            a. Example: Calculate the midpoint of the rectangle.