1. Announcements
2. On to algorithms
   a. Algorithms are recipes for doing something
   b. Given a problem
      i. Find an algorithm
      ii. Convince yourself that the algorithm works
         1. We did this with flowcharting
      iii. Implement the algorithm in a programming language
   c. Simple algorithm
      i. Making a ham and cheese sandwich
         1. Inputs – 2 slices of bread, a slab of ham, a slice of cheese, possibly a tomato
         2. How to proceed?
      ii. Making a salad
         1. Inputs – tomatoes, lettuce, carrots, cheese, ...
         2. How to proceed?
   d. More complicated algorithm
      i. Credit card (Visa, MasterCard, Discovery) has 16 digits
         1. First 6 identify what kind and who issued
            a. Visa – 4xxxxx
            b. MasterCard – 51xxxxx-55xxxx
            c. Discover – 6011xx
         2. Next 9 digits give account number
            a. A billion possible choices
         3. Last digit is key to algorithm
         4. Luhn’s algorithm for quick check
            a. Write out all 16 digits
            b. Double every other digit starting with first
            c. Add any double digits (so e.g. 12 becomes 1+2 = 3)
            d. Add all numbers
            e. Will be divisible by 10
         5. Luhn’s gives a quick check to see if number was entered correctly
            a. But it can be faked easily
            b. CVV value is computed by more complex algorithm using encryption
      6. Sidebar on accuracy of Luhn’s algorithm
         a. 1 in 10 chance of a random number getting there
      ii. More about this kind of accuracy
         1. Social security numbers
            a. Last 4 digits are not unique id
            b. But there is only a 1 in 10,000 chance you would guess right
         2. Apple iPhone fingerprint
            a. Taking the information and storing a hash of it
            b. 1 in 50000 chance of allowing false entry
e. Other real world algorithms you might have seen
   i. Tax forms
      1. Line 51: Enter the smaller of line 49 and line 50
      2. Line 52: Multiply line 51 by .08
      3. Line 53: Subtract line 51 from line 49
      4. Line 54: Multiply line 53 by .1
      5. Line 55: Subtract line 47 from line 46
      6. Line 56: Subtract line 45 from line 44
      7. Etc.

3. More formal definition of an algorithm
   a. Define the input and output as pairs
      i. For each input, how is the output computed?
   b. Define the primitive steps (e.g. arithmetic operations) that can be used in defining the
      algorithm
   c. An algorithm is then a procedure (or recipe or sequence of steps) that
      i. Starts from the input
      ii. Going step by step using the allowable primitive steps computes the output
   d. Algorithm must
      i. Be precise and unambiguous
      ii. Cover all possible situations
      iii. Never get to a situation where it doesn’t know what to do next
      iv. Eventually stop and not run forever (in most situations)

4. Something that is not an algorithm
   a. Head and Shoulders shampoo
      i. Lather, rinse, repeat

5. Simple algorithms
   a. Converting from decimal to binary
      i. divide number by 2, write down remainder
      ii. use quotient as new number
      iii. repeat until number becomes zero
      iv. show remainders in reverse order
   b. Finding the average of a sequence of numbers
      i. sum = 0
      ii. for each number (from 1 to N)
      iii. add jth number to sum
      iv. average = sum / N
   c. Form of many algorithms
      i. Set up initial conditions (get started)
      ii. Repeat an operation until a condition is met
         1. This is the looping step we’ve seen before
      iii. Finish the job by cleaning up
   d. A few tasks
      i. How do you remove the jokers from a deck of cards?
      ii. How do you find the largest (or smallest) of a sequence of numbers?
iii. These are linear time algorithms
   1. Look at each input a fixed (in these cases, once) number of times
   2. Doubling the size of the input, doubles the time it takes
iv. How about finding the largest and smallest of a sequence of numbers?
e. A different type of algorithm
   i. I’m thinking of a number between 1 and N,
      1. You guess a number
      2. I tell you if you are too high or too low
      3. You continue until you guess my number
      4. How many guesses if
         a. N is 1000
         b. N is 1,000,000
         c. N is $2^n$
   ii. If we double N, how do the number of guesses change?
   iii. This is called logarithmic behavior (as opposed to linear)
   iv. Sidebar on logarithms if necessary
f. Differences between linear algorithms and logarithmic algorithms?
g. Are there other types?
i. Quadratic algorithms
   1. Run in time that looks like $n^2$
   2. Typically involve working with all pairs of inputs
   3. For example, sorting numbers
      a. Input is a collection of N numbers
         i. Find the largest number
         ii. Find the largest of the remaining numbers
         iii. Continue this process until the numbers are sorted
      b. Selection sort
         i. Finding the largest – N (actually N-1) comparisons
         ii. Finding the next largest N-2
         iii. Etc. N-1 + N-2 + N-3 + ... + 1 is $N(N-1)/2$
      c. Doubling the size of the input, doubles the number of steps
         i. So quadratic
   ii. Comparing logarithmic to linear to quadratic
      1. N=1, all take 1 (say)
      2. N =100, logarithmic takes 7, linear takes 100, quadratic takes 10,000
      3. N = 1 million, logarithmic takes 20, linear takes a million, quadratic takes a million million (a trillion)
6. Creepy or not creepy
   a. Ads on facebook
      i. Apparently the Russians spent $100K on ads, many targeted to Michigan (10,700 vote margin) and Wisconsin (22,748 vote margin)
      ii. According to ad agency
         1. It is possible to identify undecided voters
2. Placing an ad has a 1 in 2000 change of getting an undecided voter to change their vote to your candidate
3. 1000 impressions of an ad cost $2
4. So, it costs $4 to "buy" one vote
5. For $42,800 you could "buy" Michigan and for $90,992 you could "buy" Wisconsin.
6. Source: Ben Kunz, the executive vice president of marketing and content at the media agency Mediassociates through http://www.businessinsider.com/michigan-could-have-swung-in-the-2016-election-for-just-42800-2017-10

7. Returning to sorting algorithms
   a. Very common operation done on computers
      i. Are there alternatives to selection sort?
      ii. Sort/Merge
         1. Sort the first half
         2. Sort the second half
         3. Merge the two sorted lists
         4. How to sort the two halves
            a. Divide each into 2 halves
            b. Sort the first half
            c. Sort the second half
            d. Merge the two sorted lists
            e. How to sort the two halves
               i. Repeat as above
   5. So, what we do
      a. To sort 16 elements
         i. Sort 8
            1. Sort 4
               a. Sort 2
               b. Sort 2
               c. Merge the lists
            2. Sort 4
               a. Sort 2
               b. Sort 2
               c. Merge the lists
         ii. Sort 8
            1. Sort 4
               a. Sort 2
               b. Sort 2
               c. Merge the lists
            2. Sort 4
               a. Sort 2
               b. Sort 2
               c. Merge the lists
iii. Merge to get final result

6. Sort/Merge requires $n \log n$ steps
   a. If size 1 takes 1 operation,
      i. Size 100 takes about 700
         1. Less than 10,000 for selection sort
      ii. Size a million takes about 20 million
         1. Less than a trillion for selection sort

iii. Quicksort
   1. Choose the first element and find where it fits
      a. Do so by walking the list
      b. Divide the list into 2 parts
         i. Those greater than the number
         ii. Those less than the number
      c. Sort the 2 lists by repeating the process

b. Algorithms of this type are called recursive
   i. General form
      1. Divide the input (somehow) into 2 parts
      2. For each part
         a. Repeat the process (of dividing in 2) unless things get very small
      3. Merge the results for the 2 parts
   ii. Also called divide-and-conquer