HOMEWORK 9: §8.4-8.6, 8.9-8.10 DUE MARCH 30

Name: ______________________________

- Please refer to the syllabus regarding allowed collaboration on this homework assignment.
- All answers should be fully justified.
- Your homework should be neatly written on additional paper; you may attach this cover page if you would like to keep the questions attached to the answers.

(1) *No change!* Some country has coins in the amount of 4¢ and 13¢. You need to pay a vendor for some produce, but the vendor has no change. Our goal here is to determine which prices you can pay exactly.
  (a) Experiment, in as organized a way as you can, to determine which small prices you can pay exactly.
  (b) You should find that the largest amount you cannot pay exactly is 35¢. Prove that you cannot pay 35¢ exactly. (A formal proof will use a bit of set up then a proof by exhaustion. The exhaustion part can be included in your answer to the last part.)
  (c) Prove that every price that is at least 36¢ can be paid exactly. *(This should be a proof by strong induction.)*

(2) Let $\{f_n\}$ denote the Fibonacci sequence. Prove that for every $n \in \mathbb{N}$, $\gcd(f_{n+1}, f_n) = 1$. *(Think about the Euclidean algorithm and/or its ingredients. Use induction. Can you get away with ordinary induction, or do you need strong induction?)*

(3) (a) Give a recursive algorithm that takes as input a non-negative integer n and returns a set containing all binary strings of length n. Here are the operations on strings and sets you can use:
  - Initialize an empty set S (write as "`S := ∅`").
  - Use any explicit strings, e.g. λ, 0, 1, 00110101.
  - Add a string x (as an element) to a set S ("`add` x `to` S").
  - Concatenate two strings x and y ("xy").
  - Return a set ("`Return` S").
  - A looping structure that performs an operation on every string in a set S
    "`For every` x `in` S
      `// perform some sequence of steps with string` x.
    `End-for`"

  Your algorithm **must be recursive**, not just a looping structure.
  Bonus points for adding elements to the returned set in order of increasing value (e.g. 000, 001, 010, 011, 100, 101, 110, 111).
  (b) Verify that your algorithm is correct using induction. *(Depending on your algorithm, you may or may not need strong induction.)*

Induction is how the mathematician avoids yada yada'ing over the best part.
Seinfeld reference