CSCC63, Winter 2018 Assignment #1

Due: Tuesday Jan 30 at **11:00 PM**

You may work with a partner. **You must write your own solution.**
You can only have one partner. Neither you nor your partner can discuss this assignment with anyone else other than the course instructor and the TA. If you work with a partner then you must name your partner on your assignment.

Before starting this assignment, read the statement on plagiarism posted on the course web page.

Problem A: (30 pts) Construct a Turing machine that decides the language:

$$A = \{\#w : w \in \{0,1\}^* | w \text{ has at least twice as many 0's as 1's}\}.$$

In other words, the members of A consist of a # followed by a binary string containing at least twice as many zeros as ones. For example: $\#001100 \in A, \# \in A, \#11010000000 \in A, \#101001 \notin A$.

Describe how your Turing machine works in words, in details such as "the head moves one cell at a time to the right until it finds the first cell containing the letter Y". Then describe your Turing machine as formally as in Example 3.9 in Sipser.

Problem B: (15 pts) If n is a positive integer and $2^n - 1$ is a prime number, then $2^n - 1$ is called a *Mersenne prime.* Mathematicians do not know whether there are an infinite number of Mersenne primes, despite this being a well-studied problem.

Are the following sets decidable? Prove your answer.

$$X = \{n \in Z^+ | 2^n - 1 \text{ is prime } \}.$$

$$Y = \{m \in Z^+ | 2^n - 1 \text{ is composite for every } n \geq m\}.$$

(Recall that Z^+ is the set of positive integers.)

Problem C: (30 pts) Someone wrote an algorithm called 5-Search which tests whether an input string contains a sequence of five consecutive 5's. You have been asked to design your own code-analyzing algorithm to determine whether 5-Search works on a particular input string. The first two parts of this problem ask you to prove that this is impossible.

Consider the following subset of all pairs $(< P >, X)$ where P is an algorithm (i.e. Turing Machine) that takes arbitrary strings as input, and X is a string. $< P >$ is a description of P and $P(X)$ denotes P run with input X.

$$S = \{(< P >, X) | P(X) \text{ accepts if } X \text{ contains the sequence 55555 and rejects otherwise}\}$$

(a) Your boss suggests that you determine whether $(< P >, X) \in S$ as follows: First check whether X contains 55555. Then run $P(X)$ to see whether it gives the right answer. Why does this idea not work?

(b) Prove that S is not decidable by adapting the proof presented in the second lecture.

(c) Is S recognizable? Prove your answer.

Problem D: (5 pts) Write Church's Thesis in your own words.

Problem E: (15 pts)

(a) Consider the Turing Machine presented in Example 3.9 of Sipser. The initial state is q_1. Give it an input of 11#1110. So the configuration representing the situation before the first step is $q_1 11\#1110$.

Give the configurations representing each of the next 6 steps.

(b) Suppose C_i is a configuration representing the situation after step i of a Turing Machine and C_{i+1} is a configuration representing the situation after the next step. What is the largest possible number of characters on which C_i, C_{i+1} differ? Explain your answer.