Introduction to Algorithms	**Problem Set 9**
CS 4820, Spring 2018	**Due 11:59pm Thursday, April 19**

Hand in your solution electronically using CMS. Collaboration is encouraged while solving the problems, but:

1. **list the names of those with whom you collaborated;**
2. **you must write up the solutions in your own words.**

On this problem set, there are problem-specific instructions about the requirements for proving correctness and analyzing running time. The first problem requires a running time analysis but no proof of correctness, whereas the second requires a proof of termination and correctness, but no running time analysis.

(1) *(10 points)* Design a single-tape Turing machine to evaluate the "less than" relation on two natural numbers represented in binary. The input to the Turing machine is represented as a string over the alphabet $\{0, 1, >, ?\}$. The input string is always in the format $a < b?$, where each of a and b is a string of one or more binary digits, beginning with the digit 1. Your Turing machine should terminate in the "yes" state if the number represented by a in binary is strictly less than the number represented by b in binary, and it should terminate in the "no" state if the number represented by a in binary is greater than or equal to the number represented by b in binary. If the input violates the format requirements, any behavior is acceptable as long as your algorithm terminates.

Example: if the input is $11 < 100?$ the answer is "yes". If the input is $0011 < 100?$ your algorithm should terminate, but it is fine to terminate in any of the "yes", "no", or "halt" states because the first binary string does not begin with the digit 1.

Your answer must include a description, in English (with accompanying notation as needed), of the alphabet, state set, and transition rule of your Turing machine, and a few sentences explaining how the algorithm operates and how the specified states and transitions implement those operations. You may choose to include a representation of the transition rule in tabular form (similar to those represented in Section 2 of the lecture notes) if you wish, but the tabular representation of the Turing machine is *optional* whereas the human-interpretable explanation is *mandatory.*

Your solution should include an analysis of the worst-case running time, as a function of the length of the input string. *You do not need to write a proof of correctness.*

(2) *(10 points)* Let us call a Turing machine M "speedy" if it has the following property: for every input string x, the computation of M on input x terminates, and it does so after at most $|x| + 100$ steps.

Design an algorithm to decide whether a Turing machine is speedy, given a description of the Turing machine. In more detail, the input to your algorithm consists *only* of the description of a machine M (in particular the input to your algorithm does not specify any particular input string for M) and its output should be "yes" if M is speedy and "no" if M is not speedy.

In your solution you should describe the algorithm — an ordinary algorithm description is fine, no need to explain how to implement the algorithm as a Turing machine. Prove that your algorithm always terminates, and prove that it always outputs the correct answer. However, *you do not need to analyze your algorithm's running time, other than proving that it always terminates.*