Note that while each homework may be worth a different number of points, they will be scaled so that each contributes the same amount to your total course percentage.

For any problem involving the construction of a DFA, NFA, or regular expression, you must always explain your construction, and briefly argue that it is correct (unless the problem statement asks for a proof, in which case more details and more formality are called for).
1. Write a regular expression for the language \( L = \{ w \in \{0, 1\}^* \mid \text{all occurrences of the substring “00” in } w \text{ occur before any occurrence of the substring “11” in } w \} \).

2. Recall that if \( L \) is a language, we say that the string \( x \) distinguishes \( u \) from \( v \) iff \( ux \in L \) and \( vx \not\in L \), or vice-versa. If there is no string \( x \) that distinguishes \( u \) from \( v \), then we write \( u \equiv_L v \).

   (a) Prove that \( \equiv_L \) is an equivalence relation.
   (b) Prove that any DFA that accepts \( L \) cannot have fewer states than the index of \( \equiv_L \). (The index of an equivalence relation is the number of equivalence classes.)
   (c) An equivalence relation \( \equiv \) on strings is said to be right invariant iff for all strings \( u \) and \( v \), and all characters \( a \), if \( u \equiv v \) then \( ua \equiv va \). Prove that \( \equiv_L \) is right invariant for any language \( L \).
   (d) If \( L \) is a language for which \( \equiv_L \) has finite index, show that \( L \) is regular by describing a DFA that accepts \( L \). You need not prove that your DFA correctly accepts \( L \).

3. Decision problems for DFAs. In each case, argue that your algorithm is correct, and determine the running time of your algorithm.

   (a) Describe an algorithm that on input of a DFA \( M \) decides whether or not \( L(M) \) contains only even length strings.
   (b) Describe an algorithm that on input of DFAs \( M_1, M_2, \) and \( M_3 \), decides whether or not there is one and only one string \( w \) such that \( w \) is accepted by all of the \( M_i \), and no other string is accepted by all of the \( M_i \).
   (c) Describe an algorithm that on input of a DFA \( M \) decides whether or not \( M \) accepts some string whose length is not prime.

4. Let \( L = \{ ww \mid w \in \{0, 1\}^+ \} \).

   (a) Prove that \( L \) is not regular by showing that it fails the pumping lemma.
   (b) Prove that \( L \) is not regular by using the method of distinguishing suffixes.
   (c) Prove that \( L \) is not regular by applying closure properties to it and known regular languages to obtain the language \( \{0^n1^n \mid n \geq 0 \} \) which is known to be nonregular.

5. For each of the following, say whether or not the given language is regular. Argue briefly that your answer is correct in each case (a formal proof is not required, but it should be clear from your argument that you would know how to prove it if asked).

   (a) \( \{ w \in \{0, 1\}^+ \mid w = w^R \} \)
   (b) \( \{ w1w^R \mid w \in \{0, 1\}^+ \} \)
   (c) \( \{ wxw^R \mid w, x \in \{0, 1\}^+ \} \)
   (d) \( \{ w \mid ww^R \in L \} \) where \( L \) is a regular language accepted by DFA \( M \)
   (e) \( \{ a^i b^j c^k \mid \text{exactly one of } i, j, k \text{ is less than 37, exactly one is equal to 37, and exactly one is greater than 37} \} \)