Homework 4 Morally Due Mar 5 at 11:00 AM (Dead Cat March 10
THIS HOMEWORK IS THREE PAGES LONG!!!!!!!!!!!!!!!

1. (40 points)

   (a) (0 points) READ the $R(i, j, k)$ method (on the course webpage under notes) for GIVEN a DFA, produce a REGEX for the same language. NOTE that it is a DYNAMIC PROGRAMMING algorithm. (That means it's a recursion, but done from the bottom up instead of top down.)

   (b) (20 points) Write the $R(i, j, k)$ algorithm as a RECURSIVE program.

   (c) (0 points) READ up on memoization (there is a nice Wikipedia entry on it, plus it is in many algorithms texts and on the web in other places).

   (d) (20 points) Write the $R(i, j, k)$ algorithm as a MEMOIZATION program which has the benefits of both recursion and dynamic programming!

SOLUTION TO PROBLEM ONE

Omitted.

GOTO NEXT PAGE FOR MORE HOMEWORK

2. (30 points) For each of the following state if it's REGULAR or NOT REGULAR. If it's REGULAR then give a DFA or REGEX for it. If it's NOT REGULAR then prove that it's not regular. You may use the Extended Pumping Lemma and closure properties.

   Recall that $\#_a(w)$ is the number of a's in w. For this problem and forever, $\mathsf{N} = \{0, 1, 2, ...\}$.

   (a) (8 points) (Alphabet is $\{a\}$.)

   $$\{a^n a^{2n} : n \in \mathsf{N}\}$$

   (b) (8 points) (Alphabet is $\{a, b\}$.) Here, x^R denotes the reverse of a string (so $(aab)^R = baa$).

   $$\{w : w \neq w^R\}$$

   (c) (7 points) (Alphabet is $\{a\}$.)

   $$\{a^{\lceil \sqrt{n} \rceil} : n \in \mathsf{N}\}$$

   (d) (7 points) (Alphabet is $\{a, b\}$.)

   $$\{w : \#_a(w) \geq 10 \cdot \#_b(w)\}$$

   (e) (0 points, think about) (Alphabet is $\{a\}$.)

   $$\{a^{\lceil n \log(n) \rceil} : n \in \mathsf{N}\}$$

SOLUTION TO PROBLEM TWO

a) REGULAR: this is just $(aaa)^*$.

b) NOT REGULAR: Let $L = \{w : w \neq w^R\}$. Suppose that L is regular. If L is regular, then $\overline{L}$ is regular. Note that $\overline{L} = \{w : w = w^R\}$.
Claim: $\overline{L}$ is not regular.
Proof: Assume $\overline{L}$ is regular. It should be clear that $a^n b a^n$ is $\in \overline{L}$ for any natural number n. By the Extended Pumping Lemma, there is some integer m such that we can write the string $w = a^m b a^m$ as $w = xyz$ where $|xy| \leq m$ (that is, this part is contained in the first set of a's), $|y| \geq 1$, and $xy^n z \in \overline{L}$ for all n. Let

- $x = a^{n_1}$
- $y = a^{n_2}$
- $z = ba^n$

where $n_1 + n_2 = n$.
If $xyz \in L$, then $xyyz \in L$, so
$a^{n_1}a^{n_2}a^{n_2}ba^n \in L$.
But this is not a palindrome, hence this is not in $\overline{L}$, a contradiction. Thus $\overline{L}$ is not regular, which contradicts the initial supposition that L is regular, and so L must also not be regular.

c) REGULAR: This is a^*.

d) NOT REGULAR: Assume by way of contradiction that the language is regular. Applying the Extended Pumping Lemma gives us an integer m such that we can write any string w that is in the language as $w = xyz$ where $|xy| \leq m$, $|y| \geq 1$ and xy^nz is in the language for any natural number n. Now, let $w = b^m a^{10m}$. w is in the language since the number of a's is exactly 10 times the number of b's. By the Extended Pumping Lemma, let $w = xyz \in L$ where x, y, z are as follows:

- $x = b^{m_1}$
- $y = b^{m_2}$
- $z = a^{10m}$

where $m_1 + m_2 = m$. This implies that $xyyz = b^{m_1+m_2+m_2}a^{10m}$ is also in the language. However, it is not, since $m_1 + m_2 + m_2 > m$ (because y is at least 1 b long) and multiplying by 10 on both sides gives that $10(m_1 + m_2 + m_2) > 10m$. This is a contradiction, so the language must not be regular.

END OF SOLUTION TO PROBLEM TWO

3. (30 points) The alphabet for the following parts is $\{a\}$.

   (a) (8 points) Give a DFA for

$$L = \{a^n : n \equiv 0 \pmod{2019}\}$$

Draw a diagram. You can use "...". Try to make it have as few states as possible. How many states does your DFA have?

(b) (0 points, think about as it may be on the midterm) Is there a DFA in part (a) that has fewer states than yours?

(c) (8 points) Give a DFA for

$$L = \{a^n : n \not\equiv 0 \pmod{2019}\}$$

Draw a diagram. You can use "...". Try to make it have as few states as possible. How many states does your DFA have?

(d) (0 points, think about as it may be on the midterm) Is there a DFA in part (c) that has fewer states than yours?

(e) (7 points) Give an NFA for

$$L = \{a^n : n \equiv 0 \pmod{2019}\}$$

Draw a diagram. You can use "...". Try to make it have fewer states than the DFA. You may fail. How many states does your NFA have?

(f) (0 points, think about as it may be on the midterm) Is there a NFA in part (e) that has fewer states than yours?

(g) (7 points) Give an NFA for

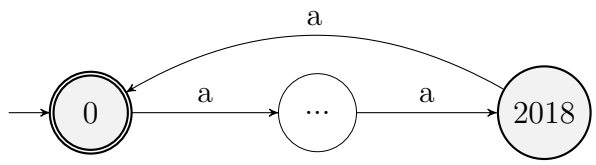
$$L = \{a^n : n \not\equiv 0 \pmod{2019}\}$$

with fewer states. Draw a diagram. You can use "...". Try to make it have fewer states than the DFA. You may fail. How many states does your NFA have?

(h) (0 points, think about as it may be on the midterm) Is there an NFA in part (g) that has fewer states than yours?

SOLUTION TO PROBLEM THREE

(a) The DFA is pictured below. The "..." state represents 2017 states with "a" transitions going right. The states are labelled $\{0, 1, \ldots, 2018\}$. If you feed in a string a^n to the DFA and it ends up in state i, then $n \equiv i \pmod{2019}$. The DFA has 2019 states.



(b) The DFA in part (a) is optimal.
Proof:
Let M be a DFA with ≤ 2018 states. Left to the reader.

(c) This is the DFA from part (a) but with the final and non-final states swapped. So this has 2019 states.

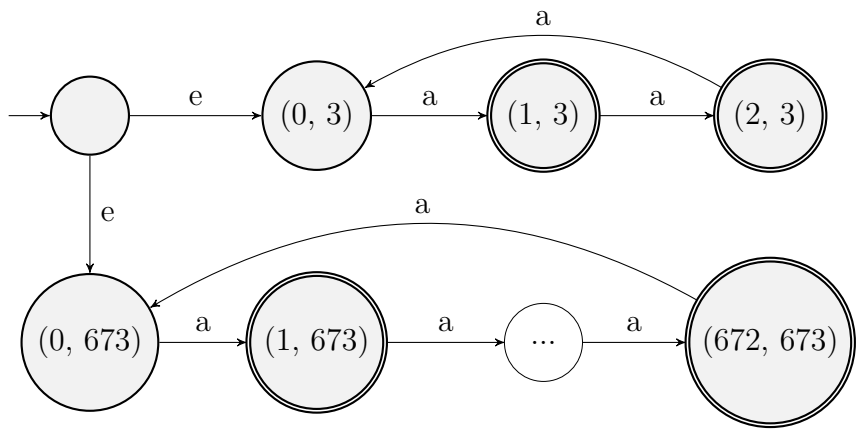
(d) The DFA in part (c) is optimal.
Proof:
Let M be a DFA with ≤ 2018 states. Then, the DFA that recognizes $\overline{L}$ also has ≤ 2018 states which contradicts part (b). So it must have 2019 states.

(e) The NFA is the same as the DFA in part (a). So it has 2019 states.

(f) The NFA in part (e) is optimal.
Proof:
Let M be an NFA with ≤ 2018 states that accepts L. Suppose the machine is run on $a^{2019} \in L$. Look at the accepting path for this string. Since the length of the string is greater than the number of states in the path, there must be a loop in the path. If we remove the loop, then the NFA accepts a string with less than 2019 a's. Contradiction, so the NFA must have 2019 states.

(g) The NFA is pictured below. The "..." state represents 670 states with "a" transitions going right. The non-start states are labelled (i, j) where

$$\Delta((i, j), a) = (i{+}1 \text{ (mod j)}, j) \text{ for } i \in \{0, ..., j{-}1\} \text{ and } j \in \{3, 673\}$$



Here is the reasoning:
$2019 = 3 \cdot 673$.

Let $n \not\equiv 0 \pmod{2019}$. Note that n CANNOT be both $\equiv 0 \pmod 3$ and $\equiv 0 \pmod{673}$. (if it was then it would be $\equiv 0 \pmod{2019}$).

Hence either

- There exists $i \in \{1, 2\}$, $n \equiv i \pmod 3$, or
- there exists $i \in \{1, \ldots, 672\}$, $n \equiv i \pmod{673}$

Hence the NFA does the following: there are e-transitions from the start state and

i. one of them goes to a DFA that accepts iff $n \not\equiv 0 \pmod 3$ (this takes 3 states) and
ii. the other goes to a DFA that accepts iff $n \not\equiv 0 \pmod{673}$ (this takes 673 states)

This NFA has $673 + 3 + 1 = 677$ states, MUCH less than 2019.

END OF SOLUTION TO PROBLEM THREE