1. (DFA → regular expression)

1.1. \( R_{11}^0 = \varepsilon \cup a \)  
    \( R_{12}^0 = b \)  
    \( R_{21}^0 = b \) 
    \( R_{22}^0 = \varepsilon \cup a \)  

\( R_{11}^1 = R_{11}^0 \cup R_{11}^0 (R_{11}^0)^* R_{11}^0 = (\varepsilon \cup a)^+ = a^* \)  
\( R_{12}^1 = R_{12}^0 \cup R_{12}^0 (R_{11}^0)^* R_{12}^0 = b \cup (\varepsilon \cup a)^+ b = a^* b \)  
\( R_{21}^1 = R_{21}^0 \cup R_{21}^0 (R_{11}^0)^* R_{11}^0 = b \cup b(\varepsilon \cup a)^+ b = b a^* \)  
\( R_{22}^1 = R_{22}^0 \cup R_{22}^0 (R_{11}^0)^* R_{12}^0 = (\varepsilon \cup a) \cup b(\varepsilon \cup a)^* b = \varepsilon \cup a \cup b a^* b \)  

\( R_{12}^2 = R_{12}^1 \cup R_{12}^1 (R_{12}^1)^* R_{12}^1 = a^* b \cup a^* b(\varepsilon \cup a \cup b a^* b)^+ = a^* b(a \cup b a^* b)^* \)  
\( \Rightarrow L = R_{12}^1 = a^* b(a \cup b a^* b)^* \)  

1.2. \( R_{11}^1 = \varepsilon \)  
    \( R_{12}^1 = a \cup b \)  
    \( R_{13}^1 = \phi \)  
    \( R_{21}^1 = \phi \)  
    \( R_{22}^1 = \varepsilon \cup a \)  
    \( R_{23}^1 = b \)  
    \( R_{31}^1 = a \)  
    \( R_{32}^1 = b \)  
    \( R_{33}^1 = \varepsilon \)  

\( R_{11}^2 = \varepsilon \)  
\( R_{12}^2 = a \cup b \)  
\( R_{13}^2 = (a \cup b) a^* b = a^+ b \cup b a^* b \)  
\( R_{21}^2 = \phi \)  
\( R_{22}^2 = a^* \)  
\( R_{23}^2 = b \cup (\varepsilon \cup a) a^* b = a^* b \)  
\( R_{31}^2 = a \)  
\( R_{32}^2 = (b \cup a a \cup a b) \cup (b \cup a a \cup a b) a^* (\varepsilon \cup a) = b a^* \cup a a^* \cup a b a^* \)  
\( R_{33}^2 = \varepsilon \cup (b \cup a a \cup a b) a^* b = \varepsilon \cup b a^* b \cup a a^* b \cup a b a^* b \)  

\( R_{11}^3 = \varepsilon \cup (a^+ b \cup b a^* b)(b a^* b \cup a a^* b \cup a b a^* b)^* (b a^* b \cup a a^* b \cup b a a^* b)^* a \)  
\( R_{13}^3 = (a^+ b \cup b a^* b)(b a^* b \cup a a^* b \cup a b a^* b)^* \)  
\( \Rightarrow L = R_{11}^3 \cup R_{13}^3 \)
2. (True or false)

2.1. False. For a counterexample, let \( L \) be any non-regular language such as \( \{0^n1^n : n \geq 0\} \). Since \( L \) is non-regular, \( \overline{L} \) (the complement of \( L \)) is also nonregular. Yet, \( L \cup \overline{L} = \Sigma^* \) is regular.

2.2. False. For a counterexample, let \( L \) be any non-regular language such as \( \{0^n1^n : n \geq 0\} \). Since \( L \) is non-regular, \( \overline{L} \) (the complement of \( L \)) is also nonregular. Yet, \( L \cap \overline{L} = \Phi \) is regular.

2.3. True. We proved in class that if a language is regular, so is its complement. This is equivalent to the statement that if the complement of a language is regular, so is that language itself.

2.4. False. Any set can be written as a (possibly infinite) union of singleton sets containing its elements. In particular, any language \( L \) can be written as a union of finite, therefore regular, languages: \( L = \bigcup_{x \in L} \{x\} \). More concretely, take our favorite nonregular language. We have \( \{0^n1^n : n \geq 0\} = \bigcup_{n=0}^{\infty}\{0^n1^n\} \).

2.5. False. If this were true, then by De Morgan’s Law the previous would also have to be true. For a concrete counterexample, let \( A_n = \{0^n1^n\} \) for every \( n \geq 0 \). Then for every \( n \), \( \overline{A_n} \) is regular. Assume, to get a contradiction, that the statement is true. Then \( \bigcap_{n=0}^{\infty}\overline{A_n} \) is regular, so that \( \bigcap_{n=0}^{\infty}\overline{A_n} \) is regular. But by De Morgan’s Law, the latter is just \( \bigcup_{n=0}^{\infty} A_n \), which we know to be nonregular, giving us our contradiction.

3. (\( L \) regular \( \Rightarrow \) \( \text{Max}(L) \) regular)

If \( M = (Q, \Sigma, \delta, q_0, F) \) is a DFA for \( L \), then the intuition to construct a DFA \( M' \) for \( \text{Max}(L) \) is as follows. If \( q_f \) is a final state of \( M \) and there is a non-empty string that drives \( M \) from \( q_f \) to a final state (possibly \( q_f \) itself), then \( q_f \) should not be a final state in \( M' \). This ensures that \( M' \) does not accept a string in \( L \) if there is a way of extending it to be another string in \( L \).

Formally, let \( M' = (Q, \Sigma, \delta, q_0, F', \) where \( F' = \{q : q \in F \text{ and } \forall x \in \Sigma^+, \hat{\delta}(q, x) \notin F\} \).

4. (\( L \) regular \( \Rightarrow \) \( \text{Cycle}(L) \) regular)

We observe that a string \( w \) is in \( \text{Cycle}(L) \) if and only if there is a way to split \( w \) into two parts: \( x_1 \) and \( x_2 \), such that there is a state \( q \) of \( L \)'s DFA \( M \) satisfying

1. \( \hat{\delta}(q, x_1) \notin F \) and
2. \( \hat{\delta}(q_0, x_2) = q \).

That is to say, a marble starting off in state \( q \) ends up in a final state of \( M \) upon consuming \( x_1 \), and a marble starting off in the initial state of \( M \) ends up in \( q \) upon consuming \( x_2 \). This suggests that the marble should keep track of three things: (1) the state of \( M \) where it started, (2) the state of \( M \) at which it currently is, and (3) whether it is consuming \( x_1 \) or \( x_2 \). Accordingly, each state of the new NFA \( M' \) will be a 3-vector \((p, q, i)\), where \( p \) and \( q \) are states of \( M \), and \( i \in \{1, 2\} \).

Formally, if \( M = (Q, \Sigma, \delta, q_0, F) \) is a DFA for \( L \), define a new NFA \( M' = (Q', \Sigma, \delta', q'_0, F') \), where

\[
Q' = Q \times Q \times \{1, 2\} \cup \{q'_0\} \text{ where } q'_0 \notin Q \times Q \times \{1, 2\}
\]

\[
F' = \{(q, q, 2) : q \in Q\}
\]

\[
\delta'(q'_0, \varepsilon) = \{(q, q, 1) : q \in Q\}
\]

\[
\delta'((p, q, 1), \varepsilon) = \{(p, q_0, 2)\} \text{ for every } q \in F
\]

\[
\delta'((p, q, i), a) = \{(p, \delta(q, a, i))\} \text{ if } a \in \Sigma
\]

By the discussion above, \( M' \) recognizes \( \text{Cycle}(L) \).
5. (Regular or not?)

5.1. \( L = \{0^n1^n : m, n \geq 0\} \).

Nonregular. Assume, to get a contradiction, that \( L \) is regular. Let \( s = 1^p0^p \), where \( p \) is the pumping length. Clearly, \( s \in L \) (for \( m = 0 \) and \( n = p \)) and \( |s| \geq p \), so let \( s = xyz \) as specified by the Pumping Lemma. Since \( |xy| \leq p \), \( y \) must lie entirely within the sequence of 1’s. Hence, \( xz = 1^p|y|0^p \) should belong to \( L \) by the Lemma, but it does not since \( p - |y| \neq p \), giving us our contradiction.

5.2. \( L = \{0^n1^n : m \text{ divides } n\} \).

Nonregular. Assume, to get a contradiction, that \( L \) is regular. Let \( s = 0^p1^p \), where \( p \) is the pumping length. Again, let \( s = xyz \) as specified by the Pumping Lemma. Since \( |xy| \leq p \), \( y \) must lie entirely within the sequence of 0’s. Hence, \( xy^2z = 0^p|y|1^p \) should belong to \( L \) by the Lemma, but does not since \( p + |y| > p \) so it certainly does not divide \( p \), giving us our contradiction.

5.3. \( L = \{xwx^R : x, w \in \{0, 1\}^* \text{ and } |x|, |w| > 0\} \).

Regular. Careful observation will reveal that a string is in \( L \) if and only if it starts and ends with the same symbol and is of length at least three. \( L \) is therefore captured by the regular expression \( 0(0 \cup 1)^+0 \cup 1(0 \cup 1)^+1 \).

5.4. \( L = \{0^p : n \geq 0\} \).

Nonregular. Assume, to get a contradiction, that \( L \) is regular. Let \( s = 0^p \), where \( p \) is the pumping length. Let \( s = xyz \) as specified by the Pumping Lemma. Then by the Lemma, \( xy^2z \in L \). However, clearly \( |xy^2z| > |xyz| = 2p \), yet \( |xy^2z| < 2^{p+1} \) since \( |y| \leq |xy| \leq p < 2^p \), so \( xy^2z \notin L \), giving us our contradiction.

5.5. \( L = \{w \in \Sigma_2^* : \text{the bottom row of } w \text{ is the reverse of the top row of } w\} \).

Nonregular. Assume, to get a contradiction, that \( L \) is regular. Let \( s = [0]^p[1]^q[0]^p \), where \( p \) is the pumping length. Let \( s = xyz \) as specified by the Pumping Lemma. Since \( |xy| < p \), \( y \) lies entirely in the first sequence of \([0]^p\)’s. Hence, \( xz = [0]^p|y|[1]^q[0]^p \) does not belong to \( L \), contradicting the Lemma.

5.6. \( L = \{0^n1^n : m, n \geq 0 \text{ and } m \neq n\} \).

Nonregular. Assume, to get a contradiction, that \( L \) is regular. Let \( R \) denote the language captured by the regular expression \( 0^*1^* \). Then \( L \cup \overline{R} \), and therefore \( L \cup \overline{R} \), must be regular since the set of all regular languages is closed under union and complementation. But the latter expression is precisely the language \( \{0^n1^n : n \geq 0\} \), which we know to be nonregular, giving us our contradiction.